Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

May Helen Robertsen Storvik

Case-Based Reasoning for Decision Support in Search and Rescue

Master's thesis in Informatics: Artificial Intelligence
Supervisor: Pinar Øzturk & Agnar Aamodt
June 2020

Photo: Hovedredningssentralen

NTNU
Norwegian University of
Science and Technology

May Helen Robertsen Storvik

Case-Based Reasoning for Decision Support in Search and Rescue

Master’s thesis in Informatics: Artificial Intelligence
Supervisor: Pinar Øzturk & Agnar Aamodt
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Hovedredningssentralen (HRS) has the responsibility for Search and Rescue (SAR) in Norway at sea, land and air. SAR services help people in distress in an effort to save lives during time-critical situations. Would it be possible to create a decision support system for helping SAR operators make quick decisions based on the situation assessment? Case-Based Reasoning (CBR) is a methodology that imitates the human problem-solving behavior by using past experiences to solve a new problem. The main objective of our research is to design and develop the retrieval process in the CBR component for predicting the hypothesis on the best action given a goal and the situation assessment.

A review of related work was performed for the SAR domain and for other domains using CBR in time-critical situations. The objective was to identify the approaches used and the different characteristics inhibited by each system. A prototype of the retrieval process was designed and developed for finding the most similar case given the situation assessment and goal.

The evaluation of the system is based on a comparison of different global similarity measures and finding the highest weighted attributes, given by a percentage, which yields the best performance for predicting the correct hypothesis. The results show promise and the prototype system is able to retrieve the most similar case to a query given a good global similarity measure.

Sammendrag

Hovedredningssentralen (HRS) har hovedansvaret i Norge for å koordinere søks- og rednings (SAR) aksjoner på sjø, land og i luft. SAR tjenester hjelper mennesker som befinner seg i en nødssituasjon ved å redde liv, hvor situasjonen ofte er tidskritisk. Vil det være mulig å lage et system som gir beslutningstøtte for å hjelpe SAR aktører med å ta kjappe avgjørelser basert på situasjonsvurderingen? Case-Based Reasoning (CBR) er en metode som imiterer menneskelig problem-løsende atferd, ved å benytte tidligere erfaringer for å løse et nytt problem. Hovedmålet med forskningen er å designe og utvikle ≪*retrieval*≫ prosessen for en CBR komponent som predikerer en hypotese om hvilken handling som er best gitt et mål og situasjonsvurderingen.

Vi har gjennomgått tidligere arbeid som er relatert til SAR domenet og for andre domener som benytter CBR i tidskritiske situasjoner. Målet var å identifisere ulike tilnærminger og de ulike karakteristikkene ved hvert system. Vi lagde en prototype av CBR komponenten som fokuserte på å finne og hente ut den mest lignende situasjonen gitt et mål og en situasjonsvurdering.

Systemet ble evaluert ved å sammenligne ulike globale vekter og ved å finne mengden attributter, gitt ved prosent, som gir best resultat ved å predikere riktig hypotese. Resultatene er positive og viser at prototypen klarer å hente ut en situasjon som har lik løsning som problemet i spørringen.

Acknowledgments

This thesis was written for the Department of Computer Science (IDI) at the Norwegian University of Science and Technology (NTNU) during autumn of 2019 and spring of 2020.

I would like to extend special thanks to my great supervisors, Pinar Øzturk and Agnar Aamodt, for their excellent contributions, help and feedback throughout this thesis. I also want to show my appreciation to Amar Jaiswal for answering my questions regarding CBR and how to use myCBR. Thanks, should also be extended to Ronny Justsen and InSoft Norway AS for answering questions regarding operations at Hovedredningssentralen (HRS) and to HRS for granting access to a Search and Rescue (SAR) report database. I would also like to thank our main contact at HRS, Per Olaf Torkildsen. At last, I would like to show my gratitude to family and friends for all their support and motivation over the past year.

May Helen R. Storvik
Trondheim, June 1, 2020

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 1, 3, 10, 17, 23

AIBN Accident Investigation Board Norway. 29, 30, 35, 37, 38, 39, 40, 77, 86, 87, 91, 93, 101

ASISA Agent System for Intelligent Situation Assessment. 18, 19, 23, 24, 86

CBR Case-Based Reasoning. 1, 2, 3, 4, 5, 10, 11, 13, 14, 17, 18, 20, 21, 22, 23, 24, 25, 27, 33, 34, 35, 37, 49, 50, 57, 59, 60, 62, 66, 69, 81, 86, 88, 89, 91, 92, 93, 94

EXITTS EXtraction of Information from Text and Telephone Source. 19

HRS Hovedredningssentralen. 1, 2, 7, 8, 14, 28, 29, 30, 34, 35, 37, 39, 40, 85, 87, 88, 89, 92, 94

HTN Hierarchical Task Network. 18, 19

JRCC Joint Rescue Coordination Centers. 7

KBS Knowledge-Based Systems. 4, 7, 10, 14, 17, 18, 23, 91

LOOCV Leave-one-out cross-validation. 57, 61, 70, 74, 76, 77, 81, 82, 89, 92, 93

MAP Mean Average Precision. 4, 46, 49, 57, 60, 69, 70, 73, 74, 77, 81, 88, 92

MBR Model-Based Reasoning. 13, 14, 15

MOB Man Over Board. 33, 37

MOUT Military Operations on Urban Terrain. 20

PEAR Person Environment Asset Reputation. 29, 35, 63

POB People on Board. 2, 29, 37, 63

RBR Rule-Based Reasoning. 13, 14, 23

RCC Rescue Co-ordination Center. 8

SA Situation Assessment. 18

SAR Search and Rescue. 1, 2, 3, 4, 7, 8, 9, 10, 14, 17, 18, 19, 23, 24, 25, 27, 29, 31, 33, 34, 35, 37, 38, 39, 40, 42, 66, 70, 85, 86, 91, 92, 94

SARPlan SAR mission Planner. 19, 23

SDK Software Development Kit. 49, 57

SMC Search and Rescue Mission Co-ordinator. 9

SRU Search and Rescue Unit. 9

Chapter 1

Introduction

1.1 Motivation

Artificial Intelligence (AI) has existed since the term was coined in 1956 [Russell and Norvig, 2010]. At the time of writing, AI solutions are easily accessible for everyday use through e.g. voice assistants like Siri, Alexa or Google Assistant. More advanced use of AI can be found in the health care, finance or transportation industry, among others.

Research into AI is considered interesting as there are still a lot of areas left to be fully explored. One of these areas is a sub-field, Case-Based Reasoning (CBR), that tries to imitate the human problem-solving behavior, as humans often tend to use past experiences to solve current tasks. Search and Rescue (SAR) is a domain where operators utilize knowledge of past experiences and lessons learned for problem solving.

Hovedredningssentralen (HRS) is responsible for SAR operations in Norway [Hovedredningssentralen, 2019]. The operations can be complex to coordinate, and the incidents can be time-critical. In 2018 there were 8507 SAR operations at land, sea and air that HRS participated in [Hovedredningssentralen, 2018a]. AI has the potential to help SAR operators with decision support during the lifespan of an incident by utilizing data HRS has accumulated of all lost person incidents since 2010. The data consists of experiences, competence and lessons learned.

The motivation is to contribute to AI research by creating a solid model and similarity measures for predictive decision support in SAR operations. An incidental consequence could be helping HRS save lives in time-critical situations.

1.2 Problem Description

The SAR domain is complex, therefore it is important to acquire a solid vocabulary and create an ontology for structuring domain knowledge. The vocabulary will be extracted through extended search and analysis of domain manuals, like the IAMSAR [2010] manual. The ontology will facilitate a common understanding among experts and scientists.

In this thesis we will create a design of an overall system to help in the planning and operational stages during an incident. These stages involve search planning, rescue planning and assistance or rescue of a person or vessel. A design of the overall system should be created to enable future work and a plan for how a final system should look.

The main objective of this thesis is to provide a decision support system for incidents at sea. When HRS is notified about an incident, actions are often recommended in order to avert the situation or ensure the safety of the People on Board (POB). The recommended action is based on a goal and the situation assessment. The situation assessment is based on information about the circumstances of the incident including information about e.g. the weather. The recommended action will be based on past experiences stored in the case base consisting of real SAR incidents, found through research. Due to anonymity, it will not be possible to access the HRS database on past incidents. A plan was made to visit HRS at Sola to verify modeling choices and get feedback from domain experts, however this was canceled due to the covid-19 pandemic.

1.3 Goals and Research Questions

Research goal A: Extract the domain terminology to use for building an ontology based on HRS material.

RQ1: *What terminology does HRS use in order to explain a situation at sea that can inform building an ontology?*

RQ2: *How can we use this terminology to build a case and an ontology?*

Research goal B: Design and develop the CBR retrieval process to create a hypothesis on the best action given a goal and the situation assessment using the developed ontology.

RQ3: *How can a case be represented and what will its content be?*

RQ4: *What cases found through research using the case representation of RQ1 will be used to populate the case base?*

RQ5: *What similarity measures will be suitable to the attributes in the case representation?*

RQ6: *How to evaluate the usefulness/quality of the system?*

1.4 Research Method

The Method of Design Science research will be followed, which includes six activities as presented by Peffers et al. [2007]. The first activity is *Problem identification and motivation* where the research problem should be presented and the value of the solution justified. This chapter has already addressed this activity through Section 1.1 and Section 1.2. The other activities involve *defining the objectives for a solution*, *designing and developing*, *demonstrate*, *evaluate* and *communicate*. We will do background research and perform a literature review in order to construct the hypotheses, where the hypotheses are represented through the research goals and questions. The research goals and questions represent the objectives of our solution. The hypotheses will be tested by designing and developing a prototype that can be evaluated. We will also demonstrate the prototype through an example run of the system. Finally, the results of the prototype will be analyzed and discussed before conclusions are drawn. This thesis will form the basis for communicating the results.

1.5 Contributions

Contributions made to the research community, include an ontology for representing SAR incidents at sea, which is generic and can be reused by other researchers. Furthermore, we have contributed to explainable AI by designing and developing the retrieval process of a CBR component. This includes creation of a case representation for representing incidents at sea and a small expandable case base for the CBR component. The main focus of this thesis has been on relative weighting of attributes in a case using global similarity measures and determining if a subset of the chosen attributes is sufficient for representing an incident. There is little research to be found on decision support systems in the SAR domain and those found address incidents involving aeroplanes. So, this thesis contributes by forming an introduction on CBR for SAR incidents at sea that is well documented and expandable.

1.6 Thesis Structure

This thesis is divided into 9 chapters.

- **Chapter 1: Introduction** - presents the motivation, problem description, research goals and questions, the research method that will be followed and contributions made to the research community. The *research goals and questions* are illustrated in Figure 1.1, which displays key aspects of the thesis content.

- **Chapter 2: Background** - presents background theory on the necessary information for understanding the SAR domain and will then compare types of Knowledge-Based Systems (KBS). A discussion on which system should be chosen based on the characteristics is also included. Looking at Figure 1.1 this is represented as *Background theory on Knowledge Based Systems and Search and Rescue domain.*

- **Chapter 3: Related Work** - performs a detailed literature review into *Related work*, also illustrated in Figure 1.1. In related work we will look at papers introducing KBS for the SAR domain and systems using CBR in time-critical situations for any domain. As the focus will be on the retrieval process, a review into a data-driven approach for finding global similarity weights and how to evaluate these measures was also performed.

- **Chapter 4: Architecture/Model** - presents the ontology containing extracted terminology from the SAR domain and will describe the architecture of the proposed CBR system. This chapter will also include *Global similarity measures and local similarity measures*, *ontology*, *information about incidents*, *case representation* and *cases stored in case base* as pictured in Figure 1.1.

- **Chapter 5: Implementation** - discuss how the measures based on the data-driven approach were re-implemented by us. This chapter will also present an algorithm (Algorithm 3) for finding the percentage of highest weighted attributes for each of the global similarity measures that gives the best results based on the Mean Average Precision (MAP) score.

- **Chapter 7: Evaluation and Results** - introduces how we will evaluate the global similarity measures. Evaluation methods and the produced results for the implemented prototype is illustrated as the last step in Figure 1.1, *test and evaluation of CBR retrieval.* Finally, we will discuss our interpretation of the results.

- **Chapter 8: Discussion** - will discuss how each of the research questions presented in this chapter has been met. Potential limitations will also be presented.

- **Chapter 9: Conclusion** - gives a conclusion based on the findings in this thesis. In addition, future work will be discussed. Future work will address limitations on the implemented system, how the prototype should be expanded to include the whole CBR cycle and how this prototype is part of a bigger proposed system design.

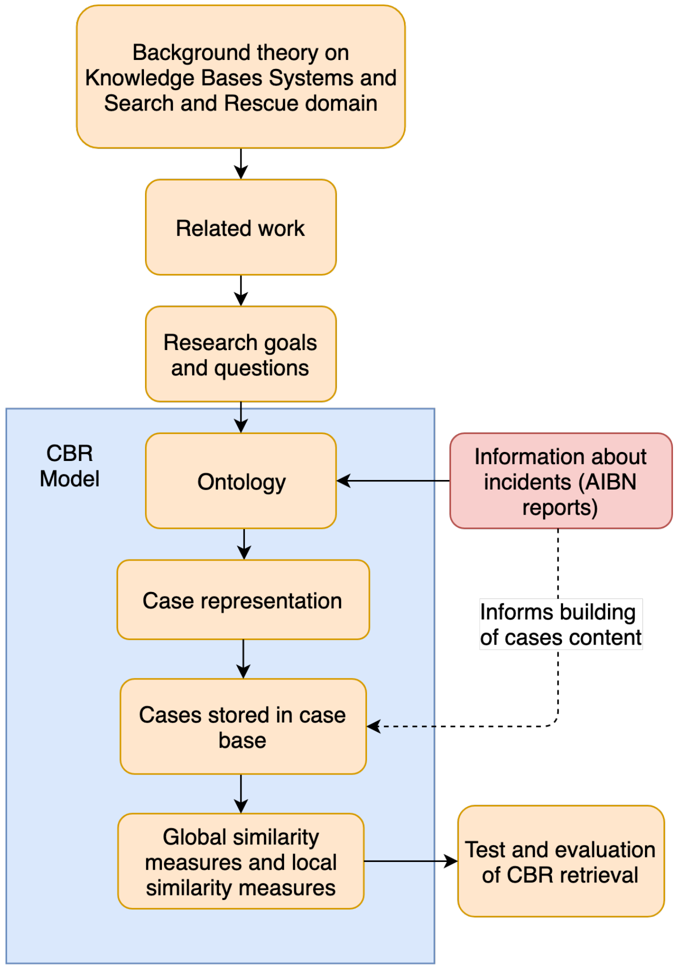

Figure 1.1: Key aspects of the content in this thesis, where the arrows inform how each component was built and the inheritance of information. The orange color symbolizes the work we have done, and the red color symbolizes how an important source has affected our work.

Chapter 2

Background

This chapter will first look into the SAR domain and the lifespan of a SAR incident in Section 2.1. Additionally, this section will also introduce background information on the Viking Sky incident. Section 2.2 will present information about building an ontology and Section 2.3 will explain similarity matrices. Finally, we will introduce three different KBS in Section 2.4 and discuss the characteristics of these in Section 2.5.

2.1 Search and Rescue

Each country needs to meet certain obligations in regard to SAR. In Norway it is HRS, which is divided into two Joint Rescue Coordination Centers (JRCC), which have the overall responsibility during search and rescue [Hovedredningssentralen, 2019]. The JRCC covers the North and South of Norway and the centers are located respectively in Bodø and Sola. This thesis will only focus on SAR incidents at sea, but aims for a generalized solution that can easily be adapted to land and air incidents as well.

SAR operators will categorize an incident into a SAR stage and an emergency phase when alerted about a situation. These categorizations provide helpful guidelines on how to search, assist or rescue given the SAR stage and the emergency phase.

2.1.1 SAR Stages

SAR operations are highly dependent on the speed at which the search or rescue is planned and carried out [IAMSAR, 2010]. A SAR incident can be categorized into different stages depending on the progress, which helps organize response activities. A SAR operation is divided into 5 stages:

- **Awareness**: is when somebody at HRS is notified about a possible emergency situation.
- **Initial Action**: is when more information is gathered and the necessary resources available to SAR are alerted. In situations that are considered urgent it might be necessary to perform additional actions from the other stages.
- **Planning**: is important for a SAR incident, especially when the location is unknown, or people move due to conditions like wind and wave current. Search planning is important so that the wrong area is not searched. Planning also encompass Rescue planning for rescue of people and final delivery of survivors to medical facilities.
- **Operations**: are all activities that involve giving assistance, search, or rescue of a missing person or vessel. The Rescue Co-ordination Center (RCC) staff will at the same time prepare for an unsuccessful search and plan subsequent searches.
- **Conclusion**: is when a person or vessel is found, or when further search is to no avail. [IAMSAR, 2010]

HRS needs quick reflexive reasoning for the Awareness and Initial Action stages, where a 20 second response time is considered too slow for a system. When authorities are alerted of an actual or potential emergency, then information gathered and initial actions are considered critical to the success of a SAR operation [IAMSAR, 2010]. A more complex system is needed for the Planning and Operations stages. Such a system would need to spend less than 20 minutes in order to be of aid in decision making. Search planning is considered the most risky, expensive and complex aspect of SAR. It is therefore important that all of the information that the SAR unit receives is analyzed and evaluated.

2.1.2 Emergency phases

According to the IAMSAR [2010] manual there exists three different emergency phases an incident might be categorized into. The emergency phases are based on the degree of concern for the safety of the people in danger. These emergency phases are:

- **Uncertainty**, which is declared when a ship fails to make a safety position report or if it is overdue at its destination. It then exists uncertainty about the safety of the ship.

- **Alert**, which is declared when a ship is not in immediate danger, but in need of help or assistance.
- **Distress**, which is when there is a high certainty that a ship is in danger and needs immediate assistance.

The emergency phases are considered important as the phases help the Search and Rescue Mission Co-ordinator (SMC) determine which actions should be performed for each incident depending on the emergency phase [IAMSAR, 2010]. Each of the phases has a checklist associated to them and the list consists mostly of information gathering actions, but there are also actions reminding the operators to dispatch a Search and Rescue Unit (SRU).

2.1.3 Viking Sky incident

Viking Sky is the incident where a cruise ship was stranded in rough sea with 1373 passengers. The incident has been used throughout this thesis, as there is a lot of public information available in news outlets and an official timeline has been posted on Hovedredningssentralen [2018b].

Viking Sky is considered a complex case due to the number of passengers and the weather conditions. The cruise ship kept drifting towards shore during a storm, thus a lot of resources were mobilized by HRS. A total of 418 people were evacuated by helicopters before the tugboat, Vivax, and the standby safety vessel, Ocean Response, started tugging the boat towards a nearby harbor. The cruise ship lost power during the incident and at one point only one out of four motors was working with no forward momentum [Hovedredningssentralen, 2018b].

The fact that the cause of the incident, engine failure and no forward momentum, was already known enabled the SAR operators to start rescue planning at once. The location was also known and thus an extended search for the vessel was not necessary.

2.2 Ontology

An ontology is considered useful if there is a need for forming domain knowledge during reasoning [Yu and Li, 2009]. It also creates a ground for common understanding among people about how information is structured [Noy et al., 2001]. The IAMSAR [2010] manual contains a lot of information and it is important to extract the correct terminology that describes an incident. It also enables explicit modeling of domain knowledge, so that it is easy for experts, in this case SAR operators, to verify the assumptions that have been made [Noy et al., 2001]. An ontology consists of concepts and the relationship between them. Each concept contains terminology represented as attributes capturing knowledge about the concept.

In order to develop the ontology, we will use a middle-out approach. A middle-out approach is in contrast to the top-down or bottom-up approach. A top-down approach will start with modeling of the top concepts building the structure through specialization, while bottom-up starts with the most specific concepts and builds a structure by generalization [El Ghosh et al., 2016]. The middle-out approach is a combination of top-down and bottom-up, so the approach includes both theoretical modeling and text analysis. The development of the ontology will therefore go out in both directions, as this is what felt most natural for the SAR domain.

2.3 Similarity Matrix

A similarity matrix is a useful tool for understanding how similar or how far apart two pair of items are according to Baxter et al. [2015] and can be referred to as a distance matrix as well. This thesis will focus on the retrieval process and a similarity matrix will quantitatively illustrate the similarity of attributes, and hence incidents. The x- and y-axis hold identical items in the same order. The similarity matrices will have a color scheme that represents a value that is the similarity between two items ranging from 0 to 1. In order to find the similarity between a pair of items a similarity function needs to be used.

2.4 Knowledge-Based Systems

According to Russell and Norvig [2010] humans know what to do based on a reasoning process that operates on an internal representation of knowledge. This form of intelligence is represented by KBS. As given by the name, the component that is central in a KBS is the knowledge base. A knowledge base contains particular information about a domain to provide “expert quality” performance in a narrowly defined area [Luger, 2008]. According to Swain [2013] the KBS is able to extend the knowledge through an inference engine or a query system. This thesis will look into the three different reasoning methods, depicted in Figure 2.1, which are considered knowledge-based methods.

2.4.1 Case-Based Reasoning

CBR is an approach to problem solving and learning that is different from many other AI approaches [Aamodt and Plaza, 1994]. CBR is an approach that uses

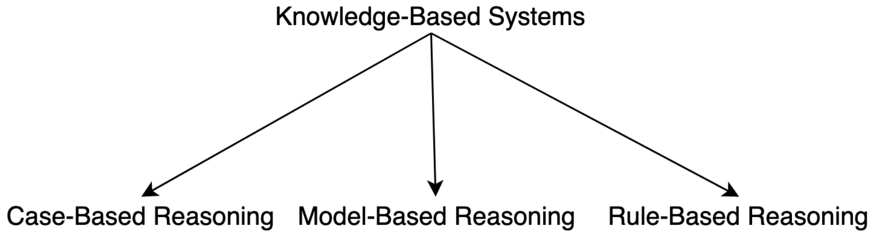

Figure 2.1: Knowledge-Based Systems

specific knowledge from past experiences that is stored as concrete cases. A new problem will be solved using CBR by finding the most similar solution among past cases, and then reusing the solution for the current problem [Richter and Weber, 2013]. An important part of CBR is the case representation, which involves figuring out what to store in a case and the structure that describes the case content. A simple way to represent a case is to use attribute-value pairs, which needs to be decided for both the problem and solution context.

After a CBR has reached a valid solution to a problem using a search-based approach, the system can retain the solution, so that if the system encounters a similar problem, search would be unnecessary [Luger, 2008]. To retain information from a new case each time a problem is solved is an approach to incremental and sustained learning [Aamodt and Plaza, 1994]. However, to achieve effective learning it is important to have a set of methods that are solid.

CBR cycle

A CBR system usually consists of the four "R"s, these are the Retrieve, Reuse, Revise and Retain processes as presented by Aamodt and Plaza [1994], see Figure 2.2.

The first process, **RETRIEVE**, can be seen in Figure 2.2. The new problem, also called the query, is used for retrieving the most similar cases or case from the case base. One of the traits in CBR is that similarity is not a general concept, but a concept that needs to differ for each case base [Richter and Weber, 2013]. The cases in the case base need to be compared to the query in a similarity assessment. The similarity assessment between two cases using attribute-value pairs consists of local and global similarity measures. Local similarity is found between single attribute-value pairs. Global similarity measures on the other hand are the relative relevance of each attribute within the whole case, defined by weights. Often the similarity score between two cases is calculated using weighted sum.

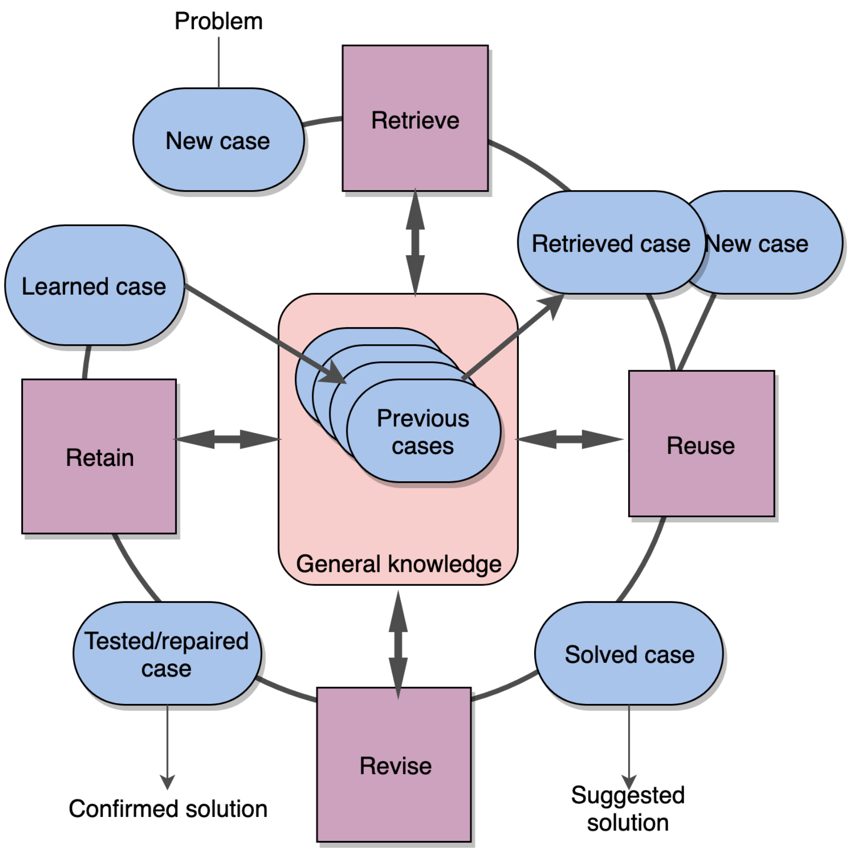

Figure 2.2: The CBR cycle illustrated (adapted from Aamodt and Plaza [1994])

A query problem should be so similar to the retrieved case or cases that the case solution can in some way **REUSE** the solution to the query. Reuse is simple if the problem situation of the query is an exact match to the retrieved case, as the solution from the retrieved case can be copied [Richter and Weber, 2013]. However, it is rarely the case that a solution can be copied directly as a solution to the query, which means it needs to be adapted.

The **REVISE** process is needed in order to determine if the adapted solution is correct, otherwise the solution should be revised. Revise will test whether the adapted query solution is successful, which can be tested in the real world or evaluated by an expert [Aamodt and Plaza, 1994]. If unsuccessful, the query solution needs to be repaired during the revise process.

If the experience gained from the query solution was deemed useful then the CBR can **RETAIN** the experience for future problem solving. The CBR learns through retaining new experiences in the case base, which makes it quicker to solve possible similar problems encountered in the future.

The general knowledge illustrated in Figure 2.2 can support the processes in the CBR cycle, but the support may vary from none to very strong. One of the aspects that distinguishes CBR from other reasoners is that it does not necessarily lead from true assumptions to true conclusions [Richter and Weber, 2013].

2.4.2 Rule-Based Reasoning

Rule-Based Reasoning (RBR) uses $preconditions \longrightarrow conclusions$ rules for representation of problem-solving knowledge in the knowledge base [Luger, 2008]. The reasoner is considered one of the oldest knowledge representation approaches for expert systems and is still a widely used approach.

The knowledge base will contain both general domain knowledge and knowledge that is case specific. A RBR consists also of an inference engine that utilizes the knowledge to find a solution for a new problem. If a new problem is applied to the RBR, then the *preconditions* that have been met will assert that the *conclusions* are true. In a RBR the knowledge is separated from control. As a result, development can be considered simpler as an iterative development cycle can be used allowing the knowledge engineer to acquire, implement and test individual rules.

A disadvantage to using a RBR is that a domain expert should provide knowledge of the problem area, as it is important to learn problem solving techniques, e.g. shortcuts or handling of imprecise data [Feret and Glasgow, 1997]. If domain experts are not available to share the skills that make them an expert, then it can be difficult for an engineer to learn and formulate such expert knowledge.

2.4.3 Model-Based Reasoning

Model-Based Reasoning (MBR) tries to address limitations to earlier rule-based expert systems where it was often the case that heuristics were applied in inappropriate situations, when a theoretic analysis should have been used [Luger, 2008]. Therefore, according to Feret and Glasgow [1997] *"a knowledge-based reasoner whose analysis is founded directly on the specification and functionality of a physical system is called a model-based system"*.

The MBR can use a mental model to represent a simplified internal representation of a domain in the real world, which allows the system to predict future outcomes [Markovits, 2012]. The model includes the critical dimensions of what should be understood in the real world and not the full complexity in order to explain deductive reasoning.

MBR can in many instances give casual explanations. Casual explanations are important if a deeper understanding of a fault is needed [Luger, 2008]. Another characteristic of a MBR is that it is considered robust and tends to be thorough and flexible in problem solving. One of the disadvantages of the MBR is if there is a lack of experiential knowledge of the domain.

2.5 Discussion

There are several advantages and disadvantages of the different KBS presented. Based on the problem description in Section 1.2 and the background theory we have presented on KBS it was decided that CBR was a good choice for decision support for the SAR domain. The fact that CBR uses knowledge from past experiences stored as concrete cases were valued, as HRS already has a system containing data on past incidents for use in the future. CBR makes it possible to reuse existing experiences and adapting these if necessary, to a new similar problem. One of the reasons that RBR was disregarded was that this method would require extensive access to experts, in contrast we were limited to data gathered from official sources. The official sources rarely convey the mental model of experts working for SAR. Furthermore, a RBR often has difficulties handling missing information [Luger, 2008]. When there is need for predicting the best action for an incident there is usually a lot of information that is missing. As information is gathered slowly over time, the action that is recommended is usually updated every time new information emerges.

CBR seemed the most promising, as SAR is a time-critical domain and if the case base contains a matching case it can usually be solved faster than with a RBR or MBR alone [Luger, 2008]. Another important factor was the fact that CBR does not rely as heavily on experts, but encourages cases to be gathered from other sources. Maintenance is also considered easier compared to that of the RBR.

MBR has some advantages that seem promising, so a hybrid system will be explored. The MBR would represent the general knowledge layer of the CBR, as seen in Figure 2.2, where the simple MBR would consist of a small ontology to represent structural knowledge of the goal and sub-goal relationship. The use of a MBR alone was disregarded due to the high complexity such models entails, as a consequence of the high level of detail that the models should capture [Luger, 2008].

Chapter 3

Related Work

This chapter will first explain the process of finding related work in Section 3.1. Next, Section 3.2 will present recent research on decision support systems for the SAR domain. Then, we will look into CBR for time-critical situations applied to any domain in Section 3.3. A paper relating to the retrieval process in CBR will also be presented in Section 3.4. Finally, there will be a discussion in Section 3.5 on the findings.

3.1 Process

For the literature review Google Scholar and Scopus were used to locate the related work. The related work includes AI research of the SAR domain, CBR solutions for time-critical situations and the retrieval process in CBR. Research relating to the SAR domain address AI-based decision support systems and not only CBR solutions. This decision was made as there is a limited amount of research on the SAR domain and thus any KBS were deemed to be of interest. CBR solutions for time-critical situations were also investigated to ensure that state of the art CBR systems were taken into consideration. It was decided to focus on time-critical situations, as SAR situations are considered time-critical for the rescue of people. In addition, a paper describing a data-driven approach for finding global similarity measures were deemed relevant, as the focus will be on the retrieval process and access to SAR operators were very limited.

3.2 Search and Rescue Domain

Research entailing decision support for aiding SAR operators were found to be limited. By performing a literature review we found that there had been conducted research for SAR services in Canada on incidents involving aeroplanes.

3.2.1 Knowledge-Based System prototype

Irène Abi-Zeid has contributed on several research projects for the search and rescue services in Canada, one of these articles focuses on a KBS prototype [Abi-Zeid et al., 2010]. The objective of the long-term research project is to aid a Canadian search mission coordinator in locating a missing aeroplane on land.

The KBS prototype takes known information about a missing aeroplane as input. The output of the system is scenarios describing what might have happened, why and where the aeroplane might be located. In addition, the system provides plausible routes followed and the possibility area of where the aeroplane might be located. The KBS system uses a rule-base, where lessons learned and knowledge have been accumulated from coordinators to be recorded in the expert system of the prototype.

In an earlier article, by Schvartz et al. [2007], the engineering of the knowledge model used in Abi-Zeid et al. [2010] is described. According to Schvartz et al. [2007], CBR was considered, but had to be disregarded due to a lack of detailed information on past cases. The knowledge model was implemented in CLIPS, a tool for creating rule-based expert systems, which made it possible for a coordinator to create scenarios on what might have occurred for the missing aeroplane to be in distress.

3.2.2 Agent System for Intelligent Situation Assessment

An earlier paper made by Abi-Zeid et al. [1999] looks more thoroughly into whether CBR is applicable to the coordination of SAR operations. The paper identifies some uses of CBR for SAR, these are online help, real time support for situation assessment and report generation. It is the study into real time support, their Agent System for Intelligent Situation Assessment (ASISA), for Situation Assessment (SA) that is most interesting for Abi-Zeid et al. [1999] and for this thesis.

SA occurs in the uncertainty and alert phases. SA is defined in the paper as the process of finding the cause of an incident, using hypothesis formulation through an information gathering process. The SAR operators discarded the proposal of using a rule-based system for aiding in decision making as the operators considered the system too "rigid". The researchers chose to go for CBR as it was confirmed by the operators that they solved new problems by making use of past similar cases.

The cases created for the ASISA system consisted of both a CBR with a hypothesis on the potential cause and outcome, and a Hierarchical Task Network

(HTN) for the information gathering process. ASISA finds first a possible hypothesis for the incident and then it finds the information gathering process to be executed from the HTN that is associated to the hypothesis. These steps form a cycle that continues until ASISA terminates with a conclusion about the nature of the incident [Abi-Zeid et al., 1999].

According to a later article by Abi-Zeid and Lamontagne [2003] the feasibility study by Abi-Zeid et al. [1999] evolved into three research projects. They were: EXtraction of Information from Text and Telephone Source (EXITTS), ASISA and SAR mission Planner (SARPlan). The implemented ASISA system gives a dynamic checklist to the coordinators with information gathering tasks to be performed and should in the future help the coordinators by automatically retrieve certain information from an incident [Abi-Zeid and Lamontagne, 2003].

An overall architecture for designing the ASISA system was created by Yang et al. [1998]. The paper includes all of the features described above in the architecture for designing the system. However, there is no paper describing the details of the implemented ASISA system and thus no results on its performance.

3.2.3 SARPlan

SARPlan is a geographic system whose objective is to help conduct mission planning and thus improve the effectiveness of response by SAR operators in air incidents occurring over land [Abi-Zeid and Lamontagne, 2003]. The system is a decision support tool for finding the optimal search strategies that maximizes the chances of success and is based on search theory [Abi-Zeid and Lamontagne, 2003; Abi-Zeid and Frost, 2005]. SARPlan has won three awards of excellence in the domain of information technology in 2001 [Abi-Zeid and Frost, 2005].

SARPlan makes it possible for a coordinator to define a possibility area, which is defined by the IAMSAR [2010] manual as "*the smallest area containing all possible survivors or search object locations which are consistent with the fact and assumptions used to form the scenario*". The final system makes it possible for a coordinator to evaluate other feasible plans than the theoretically optimal one. As a result, the coordinator or SARPlan can quickly create a search plan that is nearly optimal. The direct benefit of the system is that it helps save lives by decreasing time spent on search planning [Abi-Zeid and Frost, 2005].

3.3 CBR for Time-Critical Situations

This section will present two CBR systems used in time-critical situations and the results produced by the systems.

3.3.1 Snap: A time-critical decision-making framework for MOUT simulations

A framework created by Ting and Zhou [2008] focuses on using CBR in combination with thin slicing for making time-critical decision in uncertain situations in Military Operations on Urban Terrain (MOUT) simulations. Thin slicing is a technique that allows for quick recognition of the situation based on some key clues. The technique is similar to that of humans when presented with an uncertain situation that requires quick recognition.

Ting and Zhou [2008] decided to use CBR, as humans utilize past experiences more often in time-critical situations. Additionally, the soldiers in the MOUT simulations needed to inhibit the same tactical behavior as humans to be able to handle the complex warfare situations that arises. The CBR process retrieves past threats from a case base and proposes a solution consisting of actions to be performed. The "new case" illustrated in Figure 2.2, is also called situation assessment. For the Snap CBR component, the situation assessment gets input from a goal and an observe component. The observe component represents the environment, while the goal component is a constraint set. Together the situation assessment, goal, and observe component make up the thin slicing.

Each case is represented as $< threat, solution >$, where threat uses precondition cues to represent the data. The data is represented in qualitative measures, as a human will automatically think that a target is far away or a short distance away, instead of using e.g. meters that are a quantitative measure. The solution given by the CBR process is a sequence of actions for the soldier to perform. The soldier should e.g. hide before firing, so as not to be killed. Each action comes with post-conditions and if these are not met, the following actions will be disregarded, and the solution will have failed. The "revise" process of the CBR component is performed by experts and not the system.

The framework was integrated with the environment, called Twilight City, for testing. Ting and Zhou [2008] wanted to test soldier behavior given different experiences in the case base and various situations. The results are the average over 10 simulations. In the test there were two different soldier bots: S(A) which only had experience with Counter Strike and S(B) which had the Hasty Attack and

Retrograde experiences stored. These soldier bots were compared to Unreal Tournament bots (UT) which are programmed with default tactics in UT.
The S(A) soldier and UT bots did not handle the *Close Combat* situation logically, by exposing themselves to danger, as neither had the necessary experience. However, the S(B) soldier did manage to handle the situation by using what is called the Retrograde experience, which is closer to how a real soldier would have acted. When the bots entered the *Sniper Assault* situation, the S(A) soldiers had the correct experience to handle the situation correctly and did so by creating a smoke screen. S(B) handled the situation less optimally, by using the Retrograde experience which was not sufficient for the current situation. The UT bots did as programmed and stood in danger by not creating a smoke screen, nor seeking cover.

A second test was conducted, here the mortality rate of UT bots was tested against soldiers with 7 experiences and soldiers with 4 experiences. The simulation lasted for 20 minutes and after 2 minutes the number of UT bots that were still in play decreased faster than the soldiers. In fact, the UT bots were all dead after 14 minutes. However, after 20 minutes the number of soldiers with 7 experiences left in the game were 8.7 out of 20, while the soldiers with 4 experiences left were 2 out of 20. These findings showed that the mortality rate is dependent on the number of experiences in the case base. Where the mortality rate decreases with more experiences.

3.3.2 Predicting real-time drilling problems and improving drilling performance

Raja et al. [2011] explores a methodology that a computer system called The DrillEdge is based on. The CBR method use real-time drilling data to retrieve past experiences to predict possible future problems. The problems need to be presented to the user within a specified time-critical window, as a problem resulting in non-productive time costs about 15-30% of total well costs [Raja et al., 2011].

In order to detect developing problems on a real-time system, the CBR component needs to constantly search through the case base for similar cases. Only when the current situation starts to look similar, above a threshold to at least one of the cases stored in the case base, is the user alerted. The problem context making up the "new case" illustrated in Figure 2.2, consists of event detection that finds symptoms in the data and context information about the well. The solution context shown to the user consists of experiences and lessons learned on how to solve the current problem. The decision about storing a suggested solution to the case base as a new experience, is made by the user after the current situation is solved.

The DrillEdge system uses the methodology described and gives a visual user interface of the results. In order to evaluate the performance of the system, DrillEdge was trained on pre-drilled data, and tested on whether the system was able to predict and diagnose problems that arose in the given data set. One of the tests was a live-test, which was considered successful for situations compromised of *stuck-pipe* problems by Raja et al. [2011]. However, when live-testing *lost-circulation* problems the current scenario significantly varied from the past stored *lost-circulation* problems in the case base, so the live-tested problems were not detected. Thus, receiving modest results for *lost-circulation* problems. The conclusion of the live test was that it is important to include the whole range of situations that might arise within a problem area for good results. Relatively few cases are needed however to include a wide set of situations in complex domains [Raja et al., 2011].

3.4 Retrieval Process in CBR

This thesis will only focus on the modeling of the case representation and the retrieval process in the CBR system. Thus, it was important to find literature entailing how to develop good global similarity measures and how these should be evaluated.

3.4.1 A data-driven approach for determining weights in global similarity functions

The paper by Jaiswal and Bach [2019] introduces a method for finding initial global similarity weights for the retrieval process in CBR. A data-driven approach is used for finding global similarity weights based on score and rank. The method takes the data set used in the case base of the CBR system as input and uses multiple feature relevance scoring methods to find the relevance of the features for each of the scoring methods. A formula is presented that calculates the global similarity weight for each of the features, also addressed as attributes.

The proposed method also takes a percentage that defines the proportion of features with the highest rank that should be selected for each of the scoring methods. This was desirable as it allowed the method to determine if all or only a subset of features was needed for a classification task. For example, 50 will choose 50% of the top ranking features for each scoring method. The final number of features depends on the percentage, the amount of feature relevance scoring methods, the relevance of a feature and the score that each feature receives from the various scoring methods. The features that are returned by each of the scoring methods receive a rank depending on the position, given a list of scores sorted in descending order. The highest scoring feature will receive a rank that is equal to the maximum number of features returned by a scoring method, where the lowest receives a rank of 1. If equal scores are encountered, then the rank of the previous feature will be given to the feature in question. After all of the features returned by the scoring methods receive a rank, the ranks for a feature are summed together and used in a formula for finding the global weight for a feature.

Experiments using the proposed method are evaluated using 10-fold cross-validation. The results are presented using confusion matrices for 10-fold cross-validation and a box plot is created of received F1-scores, which are calculated over 10 runs. The confusion matrices compare the rank similarity measure to a the measures called manual and equal. The manual measure is based on domain knowledge and the equal measure has no domain knowledge, so all weights are equal to 1. The percentage of features selected are all, 75 and 50 percent.

The results are in fulfillment of the hypothesis of the paper in that distributions and statistical relationships in a data set can be used to find initial weights for the global similarity measures [Jaiswal and Bach, 2019]. From the results it is possible to determine if a subset of features performs better in a classification task. By gradually reducing the number of features it becomes possible to create the best possible system before presenting it to a domain expert.

3.5 Discussion

There is little research available on decision support systems based on AI for the SAR domain. The SAR domain is regarded as complex, and the government of each country has the responsibility of SAR within its own borders. As a result, it is a domain where it can be difficult to get enough information and data for performing research, as most SAR operations are confidential. Additionally, access to SAR operators are needed to learn about their mental reasoning.

Section 3.2 presented the most relevant AI research found for decision support systems in the SAR domain. Abid-Zeid has authored or co-authored all of the research presented in Section 3.2 for the SAR services in Canada. The KBS prototype, see Section 3.2.1, predicts casual hypotheses, but uses a RBR due to lack of case data. Whereas SARPlan, see Section 3.2.3, helps during the planning and operations stages by finding the best search strategies. Therefore, the ASISA system, see Section 3.2.2, was considered the most relevant to look at as it addresses an architecture for designing a CBR system for predicting causal hypotheses with information gathering steps. However, we found no paper addressing details of the ASISA system in regard to case representation, similarity measures or produced results.

A literature review of CBR systems for time-critical situations was also performed. This was considered necessary, as a study of methods and results of CBR systems that tackled time-critical situations for other domains could have important findings not present in related work for the SAR domain. The framework, Snap 3.3.1, was regarded highly relevant as it discussed decision support under uncertain situations and how humans only rely on some key clues for quick recognition of a situation. During a SAR incident the information is often partial. It can be argued that military officers and SAR operators need to inhibit some of the same traits. Especially, as both rely on lessons learned and experience gained from years of service, but also the need for quick situation awareness in order to handle time-critical situations. The findings from Snap showed how the bots learned to handle situations correctly like real soldiers, given that the experiences were stored in the case base.

The testing of DrillEdge, see Section 3.3.2, gave results that were also important to regard. One of the live-tests performed moderately, as some of the situations in the live-test were too different from those stored in the case base. As a consequence, the system was not able to pick up the problem. The ability to pick up a problem is important as it needs to be handled by operators within a specified time-critical window. In the other live-test the problems were similar enough to those that were stored, giving successful results.

The research on both DrillEdge and Snap show how crucial it is to cover a wide range of situations for the CBR component to recommend a solution that is considered correct.

This thesis, as stated, focuses on decision support using CBR in predicting the action for an incident at sea based on the situation assessment and goal. This way of representing the problem context is the same as Snap, except that the situation assessment will get attribute values from other sources rather than an observe component. The literature review found no previous research that has addressed our focus explicitly. All related work for the SAR domain focused on air incidents and the ASISA system was the only system that utilized a CBR component. However, the papers addressing the ASISA system only studies the architectural design of a CBR component, see Section 3.2.2, so no results were available. The feasibility study for ASISA corroborated that SAR operators value the characteristics of a CBR system. The article by Jaiswal and Bach [2019] addressed a method for finding global similarity weights through a data-driven approach. This method will be re-implemented in order to apply the same approach to our SAR data set. We considered this method important for our CBR system, as domain experts will not be available for continuous discussions.

Chapter 4

Architecture/Model

This chapter contains contributions made to the research community. The contributions consist of an ontology for the SAR domain at sea, which will be introduced in Section 4.1. Next, Section 4.2 will present a design of an overall system. Finally, contributions involving a model of the case representation, cases for the case base and similarity measures will be presented in Section 4.3.

4.1 Ontology for SAR Domain

An ontology was created in order to facilitate a common understanding of a SAR incident and how it is structured. The ontology shows the attributes of each concept, see Figure 4.1, and the relationship between the concepts. The set of attributes that were included in the ontology contains, among others, the attributes that will be important to include in a final case representation for the case base belonging to the CBR component. The ontology was created using an iterative approach as the knowledge of the domain increased.

Iteration 1

The first iteration of the ontology contained only terminology found through initial study of the IAMSAR [2010] manual. The concepts; *incident*, *target*, *boat*, *person*, *reasoner* and *environment* were created containing some of the terminology presented in Figure 4.1. *Environment* is the only concept that has not been changed since the first iteration. The concepts *uncertainty checklist*, *alert checklist* and *distress checklist* were modeled as subsets of the emergency phase. These subsets contain the checklist associated to each of the emergency phases. The checklists consist mostly of information gathering actions.

Iteration 2

The concepts *anchor* and *motor* were added as subsets to *boat* during the second iteration. These concepts were created through careful study of the official timeline for Viking Sky [Hovedredningssentralen, 2018b]. In this iteration the focus was on finding terminology that described aspects of the situation assessment and not on the resources that were mobilized. As a result, the concepts *anchor* and *motor* were created in order to represent how Viking Sky was struggling with forward momentum. In addition, the timeline describes the amount of motors and how the anchors were used to limit the drift of the ship.

Iteration 3

In this iteration the concept, *event*, was created in order to represent terminology for different event categories used by HRS. A power-point presentation containing information about the different types of events were provided by HRS, where each event is connected to a checklist. The checklist contains actions to perform depending on the event and emergency phase an incident is categorized into. The *unexpected-event* and *technical event* are considered an overall terminology containing several sub-events. This is represented through the is-a relationship illustrated in Figure 4.1.

Iteration 4

The *resources* concept was added to the ontology during this iteration. Most of the terminology was identified through the official timeline for Viking Sky at Hovedredningssentralen [2018b]. The timeline includes a lot of information about which resources were mobilized to assist during the rescue of Viking Sky. All of the helicopters at HRS's disposal were identified through the timeline. The mobilized resources were mentioned by name, as a result it was possible to find the type of vessel using the search feature in MarineTraffic [2020]. The *others* concept, which is a subset of *resources* see Figure 4.1, contains the vessel types mobilized during the incident.

At this point, the Viking Sky incident was mapped onto the current ontology to verify if the information could be correctly represented. The mapping of Viking Sky onto the ontology proved that there were enough attributes to represent the current information, which were mostly gathered from the official timeline on Hovedredningssentralen [2018b].

Iteration 5

In this iteration a former mariner, named Harry Storvik, was contacted for an informal interview. He had 3 years of experience working as a mariner worldwide at sea. The mariner then went on to become an Offshore Installation Manager with the additional role as a first-line emergency manager. The interview was conducted in order to gain further knowledge of the domain and to understand what aspects he would consider important for an incident at sea. A lot of the terminology extracted before this iteration was also mentioned as important by the mariner. However, the mariner suggested that the *area* of an incident might contain valuable information on how to respond. If an area is trafficked, assistance from other vessels could be close by and reduce the severity of the incident. Also, some areas are more inclined to adverse weather conditions and could have a longer history of past incidents. Another aspect that he deemed important was if an incident occurred inshore or offshore. As, the consequences and the accessibility of resources might change depending on where the vessel is located at the time of distress. The concept, *area*, with associated attributes can be seen in Figure 4.1.

Another observation by the mariner was the importance of knowledge about the crew, which is represented as a concept called *People on Board (POB)* in Figure 4.1. The information he regarded as important was the training and experience of the crew, as this could highly affect how well equipped the crew were to handle an incident. Also, the nationality of the crew members were mentioned by the former mariner as important. The nationality of a crew will affect whether they are familiar in Norwegian waters and the training requirements might differ from national standards. The last terminology he introduced was PEAR, which stands for Person, Environment, Asset and Reputation in order of importance. On offshore installations, Person Environment Asset Reputation (PEAR) was used as a risk assessment of what should be prioritized. From the domain knowledge acquired it seems like HRS follows the same principle, if not stated explicitly. As, e.g. the SAR operators will try to salvage the vessel if it is more likely to bring people into safety. The environment is also regarded as important and a bare minimum of crew can be left on board a vessel, if it is considered safe in order to reduce environmental damage.

Iteration 6

After extended research it was discovered that there are official incident reports from the marine department at the Accident Investigation Board Norway (AIBN) [AIBN, 2020b]. AIBN has publicly available reports on past incidents where the objective is to determine an accident's circumstances and causes, so that safety at sea can be improved [AIBN, 2020a].

The AIBN reports were read carefully and useful terminology was extracted and mapped onto the ontology. The process of mapping existing cases onto the ontology, verified that most of the extracted terminology was correct. However, the process also gave insight into other attributes that were important to include. Such attributes included *daylight* and *category*, as each incident had a *category* that described the end result, like e.g. shipwreck. Whether an incident found place during *daylight* or not was stated in the reports, as no daylight could affect the vision of the crew. An attribute representing the *time of year* was also included, as the reports mention the date and the month, which could affect the chances of an incident happening in a given area if it was prone to bad weather conditions during e.g. the winter months. The attribute could also be useful when small recreational vessels are included in the data set in the future, as incidents at sea usually increase during the summer months according to statistics at Hovedredningssentralen [2018a]. The reports also confirmed the extracted terminology from the interview with the former mariner. All of the AIBN reports that we read included information about the experience, training and nationality of the crew. The reports also contained detailed descriptions of the area and the reports mentioned if an incident found place offshore or inshore. Some of the reports also mentioned that an area was prone to adverse weather conditions and thus had a high rate of incidents. The concept *goal* in Figure 4.2 was also created after reading the reports, as information about dialogues between HRS and the captain were described. Some of the goals that were localized contain more specific goals, also called sub-goals, represented through the is-a relationship.

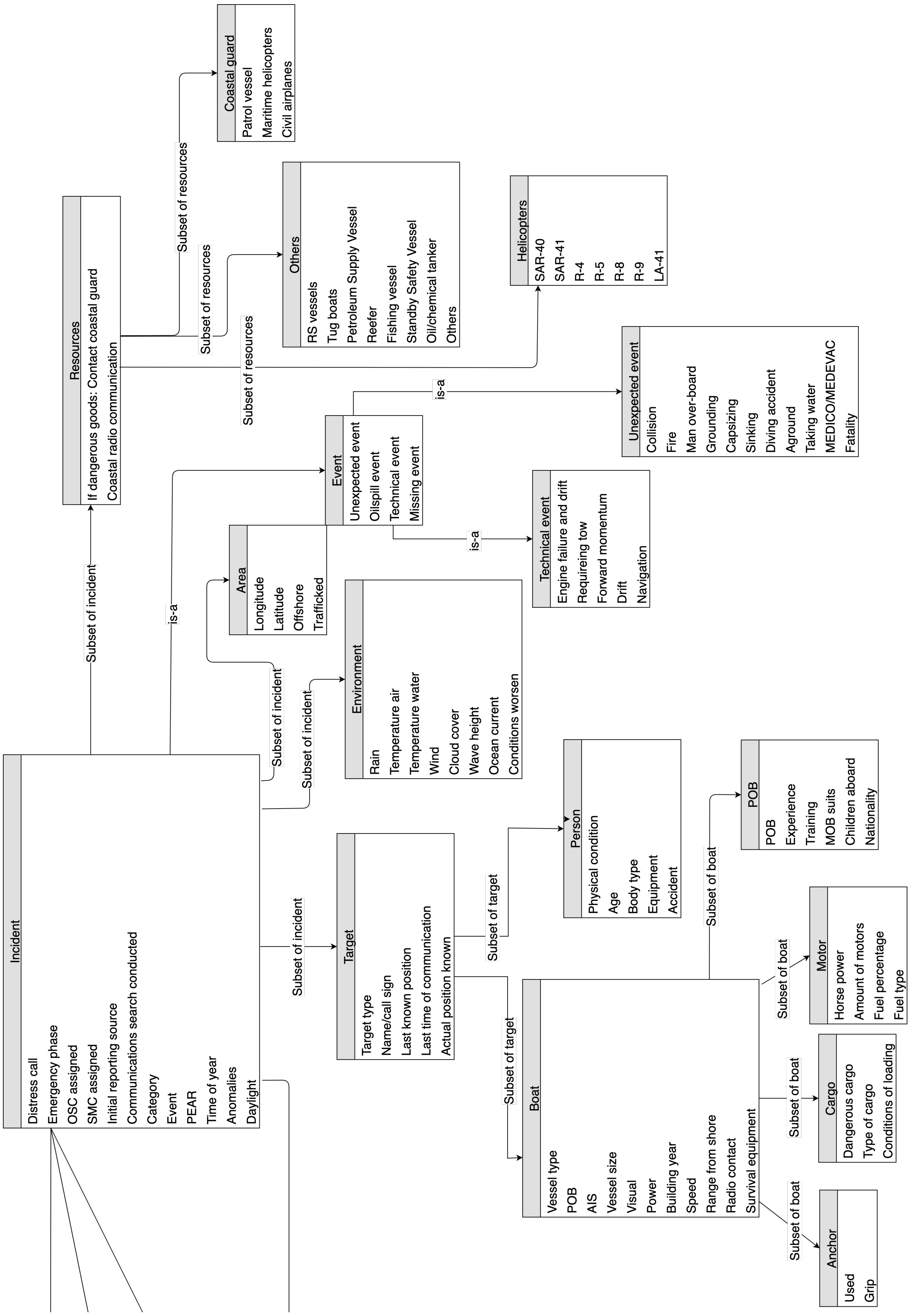

Figure 4.1: Ontology of the SAR domain

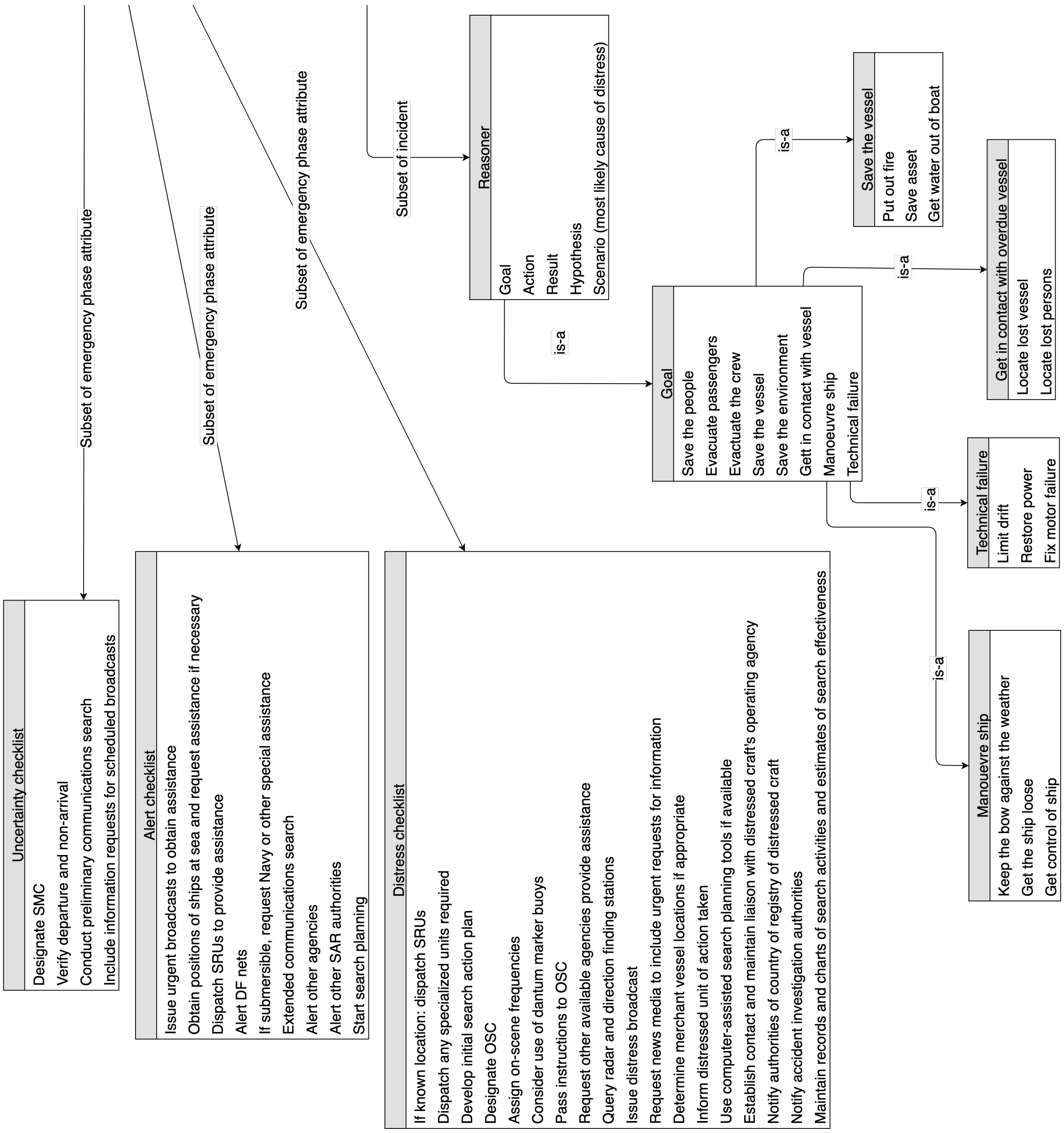

Figure 4.2: Continuation of Figure 4.1

4.2 Design of Overall System

A system design was created to show how an overall system would function spanning through the planning and operations stages shown in Figure 4.3. The complete system consists of three CBR components. Where the CBR component in Part A is considered a classification problem, and the components in Part B and C are considered planning problems. According to Richter and Weber [2013], planning problems decide on a sequence of actions to reach a goal.

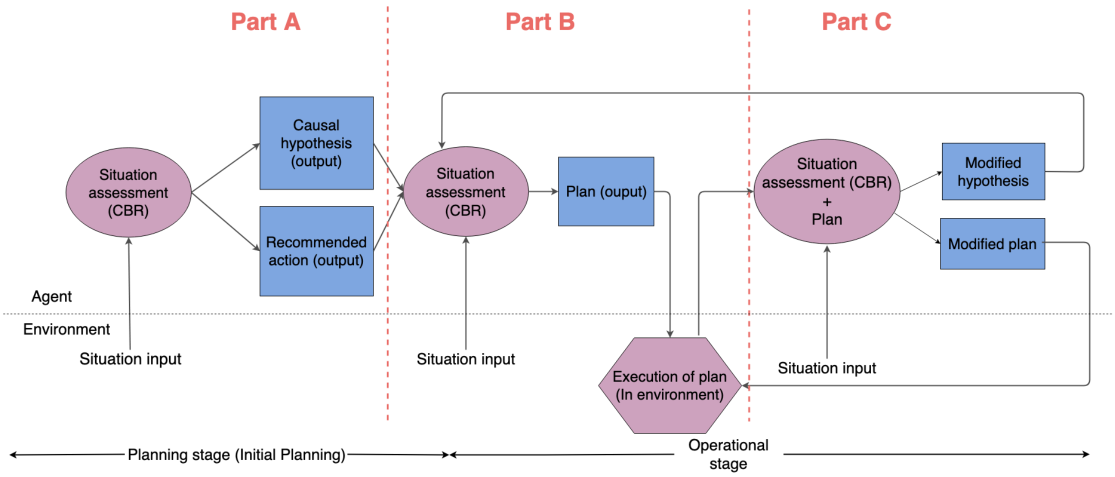

Figure 4.3: Design of complete system

This thesis will only implement the retrieval process of the CBR cycle for the *recommended action* in Part A. The *recommended action* is based on the situation assessment and a goal given by a SAR operator. If there are adverse weather conditions and it is not possible to maneuver the ship, then the goal could be to *evacuate the crew* where the recommended action could be to *take on the Man Over Board (MOB) suits*. The situation assessment needs to be correctly represented, so a good case representation must be created. The case representation for Part A, see Section 4.3.1, will be utilized in all of the CBR components. Global and local similarity measures for finding the *recommended action* are also an important part of the retrieval process.

It was valuable to design a complete system, as the implementation of *recommended action* in Part A should be developed with the complete system in mind. In addition, the overall design illustrated in Figure 4.3 should make it easier for others to continue future development of the *casual hypothesis* in Part A, in addition to Part B and Part C. The prediction of the *causal hypothesis* in Part A should use the same case representation as that of *recommended action*.

The case representation for Part B includes the case representation from Part A, extracted from the ontology, shown in Figure 4.5, in addition to a mental concept consisting of the predicted *causal hypothesis*. Part C builds on the representation of Part B, but here the plan and its results need to be added to the problem context as well. From now on the further sections will address the specific model and retrieval process for *recommended action* in Part A.

4.3 Data Modeling

The primary scope of the thesis has been data modeling, as the literature review found little contributions on modeling of the SAR domain for CBR systems. Therefore, the focus has been on creating a strong case representation and good similarity measures. Most of the modeling has been based on extended research into the domain and through the official incident reports from the marine department at AIBN [2020b]. The problem context, also referred to as the situation assessment, is represented as a set of attribute-value pairs as shown in Figure 4.4. The solution context holds the attribute-value pairs that represent the solution for a case.

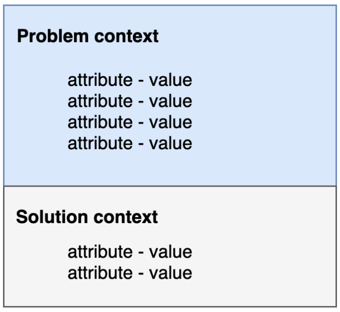

Figure 4.4: Data model of problem/solution context

Access to a HRS data base system, on reports given by mobilized resources for a specific incident, were not given until after a case representation and data set had been created. So, these reports were used to confirm attribute choices for the case representation. The HRS report system also made it possible to provide values for existing cases where a value had been unknown, or an educated guess had been made.

The domain knowledge was modeled using myCBR workbench [myCBR, 2020], and similarity measures and retrieval results were compared using the myCBR

REST API [myCBR, 2020]. myCBR is a similarity-based retrieval solution and provides a Workbench, SDK and REST API for CBR modeling and retrieval. myCBR is explained in detail in Section 5.1.

4.3.1 Case representation

The attributes chosen for the case representation were found through study of the IAMSAR [2010] manual, an informal interview with a former mariner and study of the AIBN reports. A lot of effort has been put into finding the best possible attributes for the case representation, as it contributes to a solid foundation for future development. The attributes used in the case representation are only a subset of the attributes illustrated in Figure 4.1 of the ontology. Figure 4.5 illustrates a structured view of the attributes chosen for the case representation, where the attributes are structured into subset-of concepts. A structured illustration of the attributes was deemed helpful as it gives an overview of what information the attributes describe in the real world. It also gives a clearer view of which attributes were chosen from the ontology by using the same concepts and subset-of relationships. Section 4.1 describes how the terminology for the different concepts were chosen and now we will justify why the subset of attributes were chosen for the case representation.

The following attributes up for discussion are part of the problem context, see Figure 4.4. The incident concept holds the overall information about what kind of *distress call* HRS received and what kind of *emergency phase* that the incident was categorized into. Both of these are important as HRS will act differently based on the *emergency phase* and *distress call*. *PEAR* was defined in Section 4.1 and was included in the case representation, because SAR operators might try to save the asset if it is possible without setting people in increased danger. *Time of year* was another attribute that was found interesting as weather conditions usually vary throughout the year, but also because statistics at Hovedredningssentralen [2018a] show that there are more incidents at sea during the summer months. The incident concept also includes a *category* and *event* attribute. Since, the *category* describes the end result of the incident and the *event* describes what is happening to the case at the given time. An *event* could be e.g. forward momentum, steering or drift. This leads to the attribute *goal*, which is set by a HRS operator manually to ask the CBR component to predict what *action* should be taken given the goal. The *goal* attribute is structured into a hierarchy, see Appendix A.3, with overall goals and sub-goals that can be given.

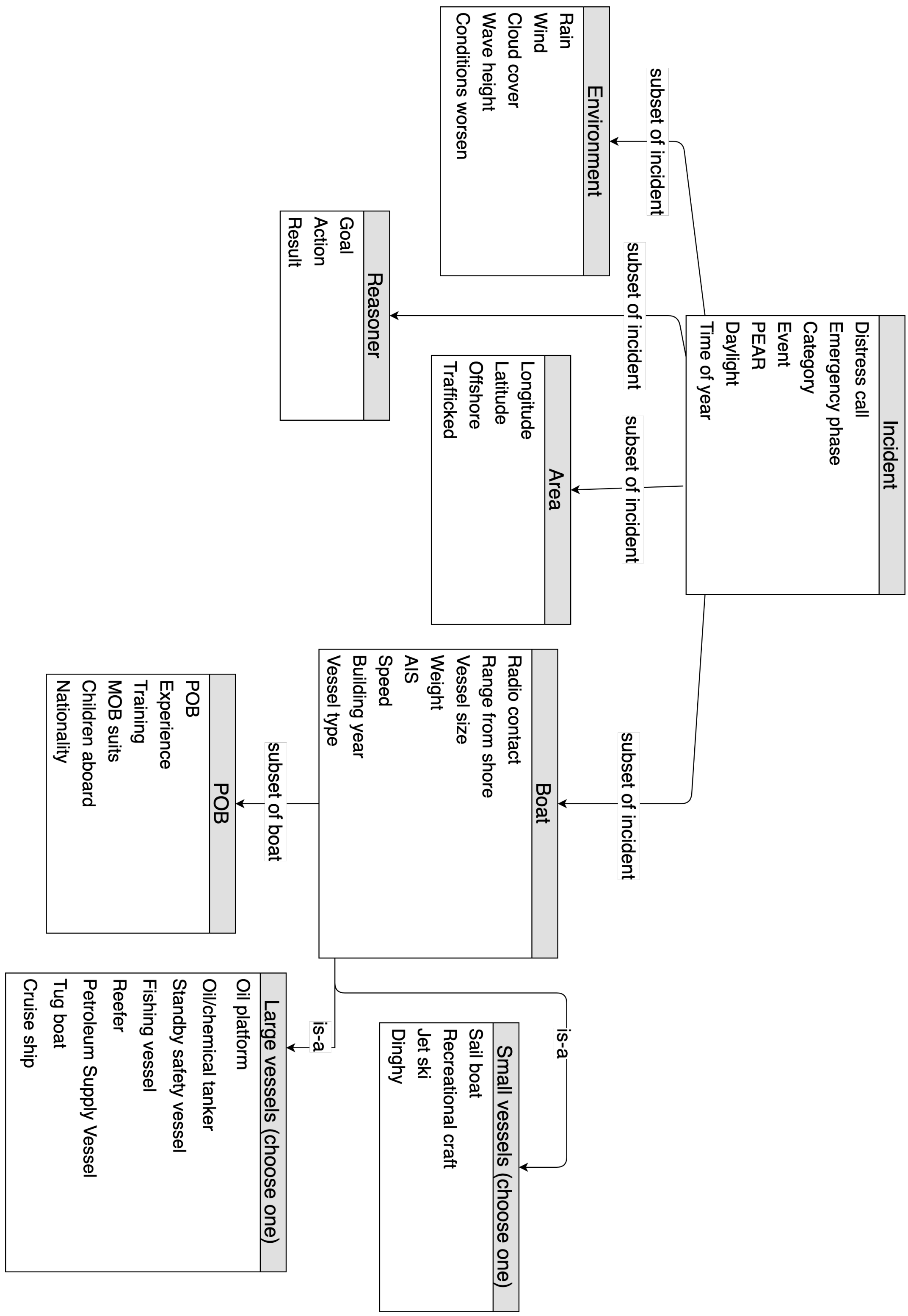

Figure 4.5: Structured view of the chosen case representation attributes

The area concept was found to be important as some places are known to be difficult waters, like e.g. Hustadvika which is known to be dangerous due to the shallow waters and reefs. Consequently, there has been a lot of shipwrecks in the area. In order to represent an area, it was decided to include the *longitude* and *latitude* attributes. Since, the prototype CBR system is not connected to any maps or traffic data it was found important to include whether an incident happened *offshore* or inshore. Inshore incidents are more likely to wind up grounded than incidents that are offshore. Whether an incident occurs in a *trafficked* area or the given *range from shore*, will affect the resources that HRS should mobilize.

Environment is also an important concept, as most of the outcomes in the incident report at AIBN happened partly due to adverse weather conditions. *Wave height* and *wind* are environmental parameters that are considered highly important. Ships struggling with forward momentum might capsize or drift toward land or reefs, while fishing boats might be in danger of sinking if hatches are open and the boat is taking water. The other parameters were considered less imminent to the incident outcome, but are still important to include in a case representation.

To store information about vessel characteristics is also considered valuable, as the *vessel size* and *vessel type* can impact the outcome of an incident. If the vessel is a large and heavy bulk carrier, the environmental damage may be huge if the vessel is grounded, compared to that of a small fishing vessel. A small fishing vessel, however, is more likely to have faults due to refurbishment or to be at risk of sinking during adverse weather conditions compared to a large bulk carrier. Especially, if the *building year* of a vessel makes it old, then there is usually a higher risk of getting a technical failure or that the boat might have flaws due to the age.

The last concept that was included represents the crew/passengers as the POB concept. Attribute values registered here are the average over all the POB. There is also an attribute called *POB* for the number of people on board. *POB* is important for which resources should be mobilized, as this might depend on the number of people on board. Additionally, we regarded the attribute *children* as important, because children are highly prioritized by HRS. *Experience* and *training* of the crew are usually mentioned in the AIBN reports, as it informs about their ability to handle certain situations or weather conditions. *Nationality crew* is also important, as foreigners might not be as familiar with Norwegian waters. Whether *MOB suits* are available, can affect the survival chances of the people if a vessel needs to be abandoned, but also which action should be taken. In several of the reports the crew has been asked by HRS to put on MOB suits and jump into the ocean, as the weather conditions made it dangerous for the SAR helicopters to evacuate from the boat.

The solution context consists of the attributes *action* and the associated attribute *result*. The *action* that should be recommended is made regardless of the *result*, but the *result* will provide additional information to SAR operators.

For the modeling in myCBR Workbench, all of the attributes have been included in a flat structure, see Table 4.1. The attributes will be of either the type; Symbol, Integer, Float or Boolean. Table 4.1 contains data for two cases as they are stored in the case base and the data is based on the AIBN reports and MarineTraffic [2020]. The table also serves as an example for what kind of values each of the attributes can contain. It is also possible to set an attribute to unknown, which would have been represented as _unknown_ in the table to be correctly interpreted by myCBR. The gray colored attributes, *action* and *result*, represent the solution context. All of the other attributes make up the problem context.

Table 4.1: Example of attribute values for cases in the case base.

Attributes	**incident0**	**incident5**
category	shipwreck	technical_failure
event	steering	forward_momentum
distress_call	panpan	mayday
emergency_phase	distress	distress
pear	asset	asset
daylight	FALSE	TRUE
time_of_year	0.08	0.25
rain	12	18.4
temp_air	3	-1.6
wind	24	25
cloud_cover	TRUE	TRUE
wave_height	8.0	18.0
conditions_worsen	TRUE	TRUE
longitude	67.63	63
lattitute	14.51	7
offshore	TRUE	TRUE
trafficated	FALSE	TRUE
range_from_shore	short	short
radio_contact	TRUE	TRUE
vessel_size	57.26	228
weight	969	4826
ais	TRUE	TRUE

Table 4.1 continued from previous page

building_year	1952	2017
vessel_type	cargo_ship	cruise_ship
speed	8	5
pob	6	1373
experience	experienced	none
training	trained	none
mob_suit	TRUE	FALSE
children_aboard	FALSE	FALSE
nationality_crew	norwegian	mixture
goal	get_control_of_ship	fix_motor_failure
action	shift_from_autopilot_ to_manual_control	check_engine_ room_for_problems
result	failed	failed

4.3.2 Incidents stored in the case base

The incidents stored in the case base have been collected by reading the AIBN reports. The reports vary in detail, but most of the incidents included enough information to make educated guesses for the attributes where information was lacking in the reports, see Appendix A.2 for cases stored in the case base. For details about the area and characteristics of shipwrecked vessels MarineTraffic [2020] was used. Appendix A.1 shows details about the incidents found in the AIBN reports, where the educated guesses are illustrated. The educated guesses are mainly made on the environmental attributes or the emergency phase that HRS categorized an incident into. This is because there was limited data available publicly on the internal workings of HRS.

At the end of the research phase for this thesis access to one of the HRS report databases was granted. The database contained reports from mobilized resources, but the reports did not provide any new information for the incidents. However, the SAR reports did confirm resource use and reassured that the educated guesses on environmental variables were correct. The database contains only cases from 2010, so it was not possible to check every case in Appendix A.1, as some of them are older.

The AIBN reports mentioned some *goals* and *actions* made for each of the incidents with an associated *result*. The *goal* attribute became part of the problem context, while *action* and *result* are part of the solution context. One of the limits with only basing the cases on the AIBN reports is the fact that these incidents were dire and the vessel involved usually ended up shipwrecked or grounded, see Appendix A.1. As a consequence, actions made to achieve a goal like e.g.

get_control_of_ship failed. The *result* attribute associated to an *action* is therefore treated as additional information. The most similar case does not need to have a successful *result* in order to be considered correctly classified.

Every incident created from the AIBN reports had multiple *goals* and *actions* described. The incidents unfold over time and different events could trigger different *goals* and *actions*. It was therefore decided that a case should be stored accordingly, so that an incident can have multiple entries in the case base where it is primarily the *action* attribute that is changed for each entry. When each incident had been split according to what affected each *action* to be triggered, it was decided to choose only the cases containing the *goals*: *evacuate_the_crew*, *limit_drift*, *get_control_of_ship* and *fix_motor_failure*, as these goals had at least 3 entries. The case base contained a total of 20 cases for the duration of this thesis.

A self-similarity matrix for the incidents stored in the case base can be seen in Figure 4.6 using the global similarity measure called *manual_wt_all*. The measure, *manual_wt_all*, has been created based on the domain knowledge we have acquired during study for this thesis and will be presented in more detail in Section 4.3.3. All of the attributes belonging to the problem context have received a global weight equal to or larger than 1 for *manual_wt_all*. Looking at Table 4.1 all of the attributes except those that are colored gray, *action* and *result*, are part of the problem context. The solution context is not included in similarity comparisons between cases and thus receives a weight of 0. Some of the cases in Figure 4.6 will only differentiate on the attribute *action*. As a result, some cases will end up with a similarity score of 1 as the problem context is equal and only differentiate on the solution context.

4.3.3 Similarity measures

Finding good similarity measures have been an important part of this thesis. Domain knowledge had to be acquired through research and was not supplied by HRS. Therefore, a lot of time and effort have been spent on finding good similarity measures for each of the attributes. Local similarity is applied to the domains of attributes and is used to compare individual attributes values through a distance function [Richter and Weber, 2013]. Each of the attributes in a query case is compared to the attributes of a case stored in the case base. If one of the attributes in a comparison between the query and case receives a local similarity score of 0, then it means that this attribute makes no contribution to the global similarity. The global similarity measures compare the whole objects or whole cases [Richter and Weber, 2013]. Each attribute is given a global weight that represents the relevance of an attribute and the amalgamation function, weighted sum, will be used in this thesis to find the similarity score of two cases for the SAR domain. The local and global similarity measures have been modeled using myCBR Workbench,

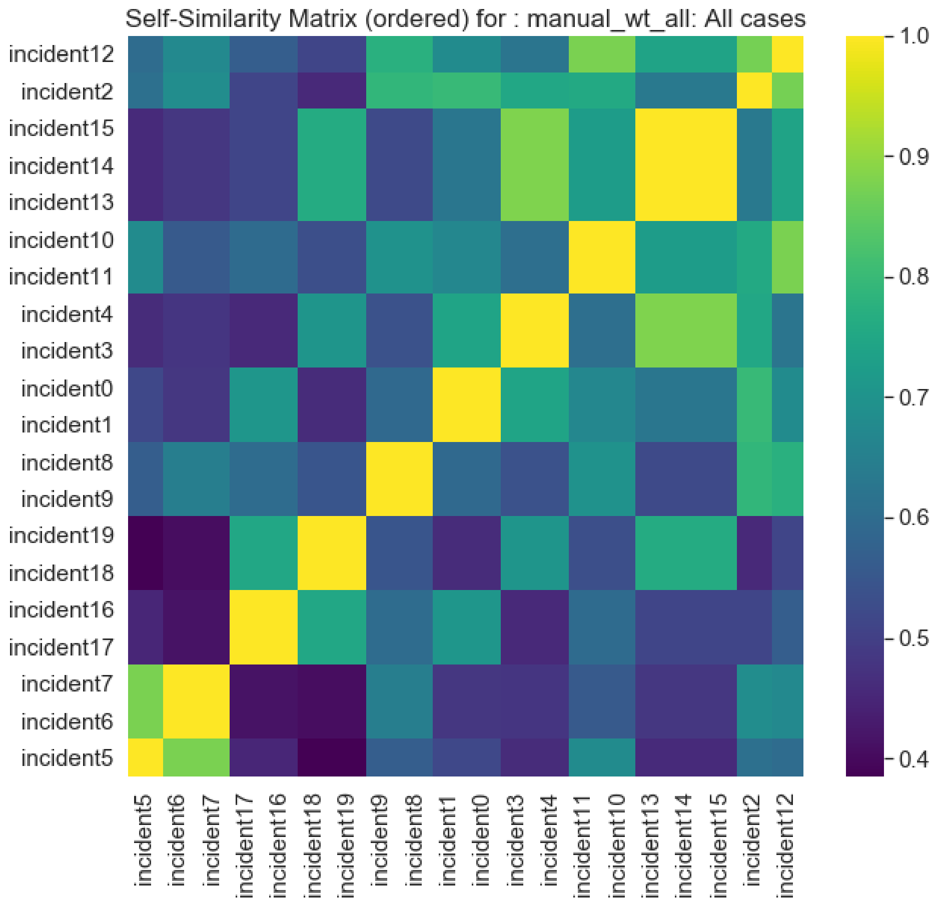

Figure 4.6: Self-similarity matrix for all cases

see Section 5.1 for more information.

Let i represent an attribute in the case representation and (C_a, C_b) be two different cases. Then we can find the contribution to the global similarity from a single attribute by

$$Global\ similarity_i\ (C_a, C_b) = global\ weight_i \cdot local\ similarity_i(C_a, C_b) \quad (4.1)$$

The total similarity score between the two cases uses weighted sum and is found by

$$Similarity\ score\ (C_a, C_b) = \frac{1}{\sum_i global\ weight_i} \sum_i Global\ similarity_i(C_a, C_b) \tag{4.2}$$

Local similarity measures

The local similarity function for each attribute has been based on the knowledge we have obtained through study and not on expert knowledge. myCBR allows for several different attribute representations, but only a few were used during the modeling of a SAR incident. All local similarity functions are included in Appendix A.4.

The local similarity for **the Integer or Float attributes** were all considered symmetric and the distance functions used were the arithmetic difference between the case and the query. The local similarity for all of the integer and float attributes are using polynomial with X, where X is the degree of the polynomial graph, to calculate the distance. Figure 4.7 shows the local similarity function for the *wind* attribute, where *polynomial with 5* is used to capture the desired distance function. Looking at the graph we can see that the maximum value for the *wind* attribute is 50 and the minimum 0. If the wind for $C_a = 24$ and $C_b = 26.7$, then using the local similarity function in Figure 4.7 will yield $local\ similarity_{wind}(C_a, C_b) = 0.758$. The degree of the polynomial graph will vary between 1 and 10 for the other attributes. A higher polynomial degree means that there is a higher sensitivity as the graph becomes steeper, so in order to receive a high similarity score two attribute values must have a small difference. One of the attributes with a high polynomial degree is the *longitude* attribute. *Longitude* for the whole world can be between -180 degrees to 180 degrees, which are also the configured maximum and minimum values for the attribute in myCBR, while in Norway the longitude is between around 58 to 72 degrees. Therefore, a high degree of the polynomial function was needed to be able to distinguish between distances in Norway when using -180 degrees and 180 degrees as minimum and maximum values.

The Boolean attributes have the default setting for local similarity measures in myCBR. This means that if the query is False and the case is False, then there is a similarity of 1. If e.g. the case is False while the query is True, then the similarity is 0.

For **the Symbol attributes** the local similarity functions in myCBR called Taxonomy function and Symbol function were used. All of the local similarity

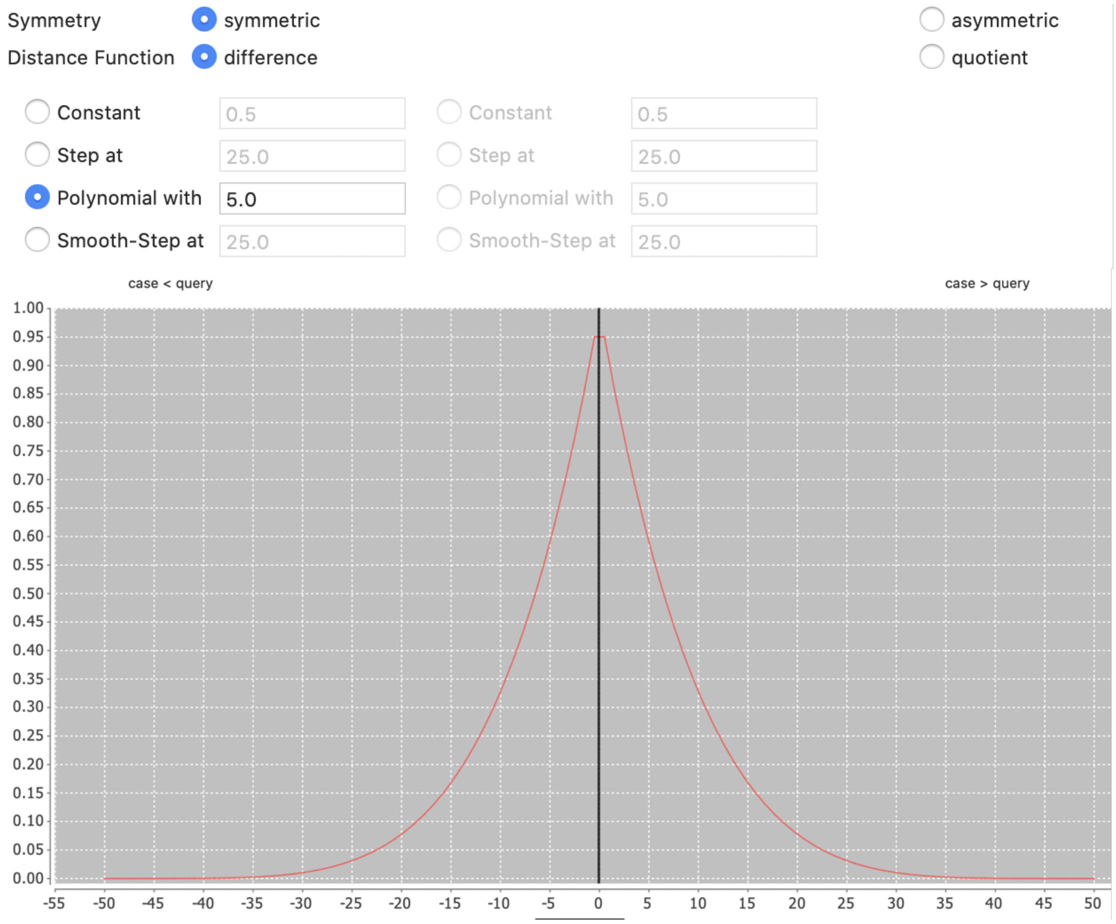

Figure 4.7: Local similarity function represented in myCBR for the Wind attribute

functions for the symbol attributes are considered symmetric and we set the local similarity functions manually at the beginning of the data modeling. As can be seen in Figure 4.8, the symbol function is a table of all possible attribute values that the *event* attribute can have. All local similarity functions have been based on the domain knowledge we have acquired through study. The values *drift* and *forward_momentum* for the *event* attribute received a local similarity score of 0.6, as the lack of forward momentum will cause the vessel to start drifting during bad weather conditions. Another *event* pair with a similarity of 0.6 is *grounding* and *shipwreck*. Both of these will result in damage or loss of the vessel, where *shipwreck* is more likely offshore and *grounding* is more likely inshore.

A taxonomy function can be seen in Figure 4.9 for the *goal* attribute. The taxonomy has structured the information into overall goals and sub-goals. A taxonomy can be used when nodes on the same level are disjoint sets and leaf nodes

	drift	forward_moment	grounding	shipwreck	steering	technical_failure
drift	1.0	0.6	0.2	0.2	0.5	0.6
forward_moment	0.6	1.0	0.2	0.2	0.6	0.6
grounding	0.2	0.2	1.0	0.6	0.2	0.2
shipwreck	0.2	0.2	0.6	1.0	0.2	0.2
steering	0.5	0.6	0.2	0.2	1.0	0.6
technical_failure	0.6	0.6	0.2	0.2	0.6	1.0

Figure 4.8: Local similarity function for the *event* attribute

are real world objects [myCBR, 2020]. The leaf nodes in Figure 4.9 have a similarity score of 1 if there is an exact match for the attribute of a query and a case in the case base. The semantic of the inner nodes was set to *uncertain*, which means that a query with the goal, *technical failure*, is looking for exactly one of the leaf nodes, but is uncertain which is correct. When the semantic of inner nodes is set to *uncertain* one needs to decide on three different methods for computing similarity between two taxonomy symbol attributes. These are pessimistic, optimistic and average. For the *goal* attribute it was chosen to use the method pessimistic, which uses a lower bound [myCBR, 2020]. If a case base has two cases with goals C_a= *evacuate_the_crew* and C_b= *fix_motor_failure*, then a query where *q=limit_drift* should receive a higher local similarity for C_b than C_a. This is due to the fact that C_b is closer to q, with *goal* at the same inner node. The taxonomy presented in Figure 4.9 should be expanded when more cases containing different *goals* are gathered. The taxonomy is built upon the goals and sub-goals illustrated in Appendix A.3.

Global similarity measures

To be able to evaluate the quality of retrieval it was decided to create four different global similarity measures. All of the measures are using weighted sum, see Equation (4.2). One of the global similarity measures we have created is called **equal** where all attributes receive an equal weight. **Equal** creates a baseline where no domain knowledge has been captured and all attributes have received a weight of 1. Modeling of the other global similarity measures should therefore perform better than **equal**. All of the global similarity measures use the local similarity functions we have created for each attribute, see Equation (4.1).

The paper by Jaiswal and Bach [2019] addresses a *"data driven approach for determining weights in global similarity functions"*, as introduced in Section 3.4.1. The paper presents an algorithm for discovering initial global similarity weights. According to Jaiswal and Bach [2019] the data driven approach is an advantage

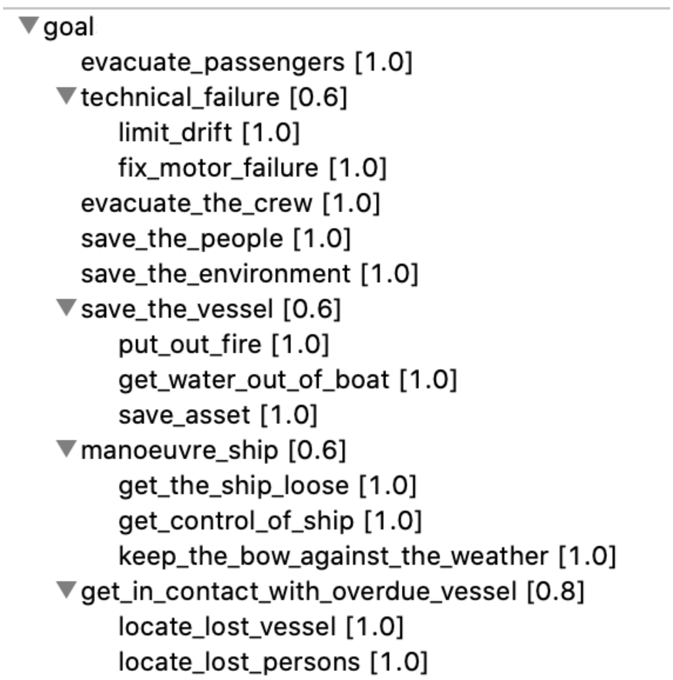

Figure 4.9: Taxonomy for the Goal symbol attribute for finding the local similarity between two cases

in absence of domain knowledge and enables the developer to discuss the setup with domain experts. As a result, we decided that it was valuable to re-implement the approach described in the paper. The paper also compares the percentage of activated attributes using all, 75% and 50% of the attributes.

The data driven approach for finding the global similarity measures is called **rank**. Each feature relevance scoring method, also called scoring methods, returns a percentage of the highest scoring attributes. The returned attributes receives a rank according to its score and position. The global weight for an attribute will be summed based on the formula on line 19 in Algorithm 1 (Section 5.3). The algorithm is a re-implementation of the one presented by Jaiswal and Bach [2019] and the details will be discussed in the next chapter. The global weights for each of the attributes can be seen in Figure 4.10. For the *incident_id* and the solution context attributes, *action* and *result*, the discriminant field is set to false and the weight to zero. The discriminant field is set to false and the weight to zero for the attributes that do not belong to the problem context. This enables a query to only regard attributes belonging to the problem context. The **rank** measure has been able to spot some of the trends in the data, where the *goal* attribute has the biggest impact on case retrieval, as it is strongly coupled with the recommended action.

The paper by Jaiswal and Bach [2019] also briefly mentions a **score** measure that uses the actual scores instead of rank when summed together into an attribute weight. However, the approach we have decided to use for deciding percentages for the **score** measure is slightly different than that of the rank measure. Instead of choosing the percentage of the highest scoring attributes the measure takes all of the attribute scores. After all the scores for each of the scoring methods have been summed together, the percentage of the highest weighted attributes will be selected. This is illustrated in Algorithm 2 (Section 5.3).

The last global similarity measure was named **manual**, as the weights for each of the attributes have been based on research and the domain knowledge we have collected through study. The **manual** measure should perform better than the baseline in order to prove that the domain knowledge is understood and that a good similarity measure has been found. Figure 4.11 shows the weights chosen for the *manual_wt_all* measure. *Goal* is weighted highest as we deemed this attribute to have the highest impact on which action should be chosen. In addition, the *event* attribute was considered to be of high relevance as the event affects what action should be recommended. The environment attributes, *wave_height* and *wind*, were given a similarity score of 7 as these were considered the most likely aspects of the environment to affect the outcome of an incident. *Longitude* and *latitude* have both received a similarity score of 8, in order to represent the importance an area can indicate if incidents occur in close proximity.

In Section 5.5 we will present an algorithm we have created for finding the number of attributes, given by a percentage, that makes each of the measures perform best according to the MAP.

Attribute	Discriminant	Weight
action	false	0.0
ais	true	2.0163934
building_year	true	21.836065
category	true	14.721312
children_aboard	true	1.0
cloud_cover	true	18.617487
conditions_worsen	true	11.333333
daylight	true	5.7431693
distress_call	true	14.382514
emergency_phase	true	16.5847
event	true	31.153006
experience	true	9.978142
goal	true	32.0
incident_id	false	0.0
lattitute	true	23.021858
longitude	true	19.125683
mob_suit	true	7.9453554
nationality_crew	true	17.93989
offshore	true	6.420765
pear	true	20.650272
pob	true	22.344263
radio_contact	true	15.398907
rain	true	18.617487
range_from_shore	true	3.0327868
result	false	0.0
speed	true	29.628416
temp_air	true	16.076502
time_of_year	true	15.737705
trafficated	true	6.420765
training	true	9.8087435
vessel_size	true	22.344263
vessel_type	true	17.431694
wave_height	true	19.295082
weight	true	23.699453
wind	true	26.07104

Figure 4.10: Global similarity weights for rank_wt_all

Attribute	Discriminant	Weight
action	false	0.0
ais	true	1.0
building_year	true	1.0
category	true	3.0
children_aboard	true	3.0
cloud_cover	true	1.0
conditions_worsen	true	1.0
daylight	true	1.0
distress_call	true	3.0
emergency_phase	true	2.0
event	true	15.0
experience	true	4.0
goal	true	20.0
incident_id	false	0.0
lattitute	true	8.0
longitude	true	8.0
mob_suit	true	6.0
nationality_crew	true	2.0
offshore	true	4.0
pear	true	8.0
pob	true	6.0
radio_contact	true	1.0
rain	true	1.0
range_from_shore	true	6.0
result	false	0.0
speed	true	3.0
temp_air	true	1.0
time_of_year	true	4.0
trafficated	true	3.0
training	true	4.0
vessel_size	true	2.0
vessel_type	true	5.0
wave_height	true	7.0
weight	true	1.0
wind	true	7.0

Figure 4.11: Global similarity weights for manual_wt_all

Chapter 5

Implementation

This chapter will first explain the tool myCBR in Section 5.1. Section 5.2 will give an overview of the implemented retrieval process for the prototype CBR component. Further on, we will have a look at the implementation of the global similarity measures "Rank" and "Score" in Section 5.3. Section 5.4 will discuss an endpoint we have created for the myCBR REST API. Lastly, we will introduce an algorithm that finds the best percentage of activated attributes for all of the global similarity measures given the MAP in Section 5.5.

5.1 myCBR

myCBR is an open-source CBR tool with focus on similarity-based retrieval [myCBR, 2020]. As a part of the research questions for this thesis we will create a prototype for the CBR retrieval process using the myCBR tool. myCBR consists of a Workbench, REST API and Software Development Kit (SDK). All of these will be utilized in different stages of creating the CBR prototype system. The workbench was used for retaining cases in the case base, configuring restrictions on possible values for the different attributes and creating local similarity functions. Global similarity measures can be created using the workbench, but it was quicker to test different global similarity measures by creating an API endpoint presented in Section 5.4. myCBR Workbench saves all configurations of the project as a .prj file, which will be referred to as aisar.prj in the next sections. The myCBR REST API is run upon the aisar.prj file that contains the CBR component and all cases stored in the case base. The myCBR REST API endpoints use functions defined in the SDK for querying aisar.prj. The myCBR team has also created a myCBR Wrapper written in Python, which makes it possible to easily gain access to the API when using Python as programming language.

Several global similarity measures will be created and so a naming convention is used. The global measures are called *manual_wt_*, rank_wt_*, score_wt_** or *equal_wt_**. The first name specifies the type of measure that is used as presented in Section 4.3.3, while *wt* specifies that the type of measure uses weighted sum for calculating the global similarity score. The * will be a number between 0 and 100, which represents the percentage of attributes that have been included or activated for each of the measures. For *manual, equal* and *score* the percentage of activated attributes will always be the same. There are 32 attributes in the case representation and if 50% is specified as the number of attributes to be included, then there will be 16 active attributes. The percentage of attributes that will be included for the *rank* measure differ slightly and will be specified in detail in Section 5.3. If only 50% of the attributes will be activated, then the global similarity weight for attributes that should not be active will be set to 0 and the discriminant to false.

5.2 Overview

An overview of the system components is illustrated in Figure 5.1. The backend consists of the aisar.prj file, that represents the CBR component, and the myCBR REST API. The case base in aisar.prj was populated using the *.csv file with cases* illustrated in Figure 5.1, see Appendix A.2 for the cases.

Jupyter Notebook is an open source web application which enables creation of documents with live code and visualizations [Jupyter, 2020]. The notebook is considered an interface as it easily illustrates textual or visual results of running a code snippet. All of the algorithms and evaluation methods that we have implemented are located in the Python middleware. Jupyter notebook will access these functions and illustrate the results for the user. The Python middleware also contains the myCBR Python API wrapper which makes it easier to access the myCBR REST API when python is used.

The myCBR REST API repository also contained an example Jupyter Notebook with example code for creating these figures 4.6, 7.7, 7.8, 7.9.

5.3 Rank and Score Similarity Measures

The paper on *Relevance-Based Feature Weighting Algorithm* was introduced in related work, Section 3.4.1, and was written by Jaiswal and Bach [2019]. The algorithm will be re-implemented as it was considered valuable to compare the

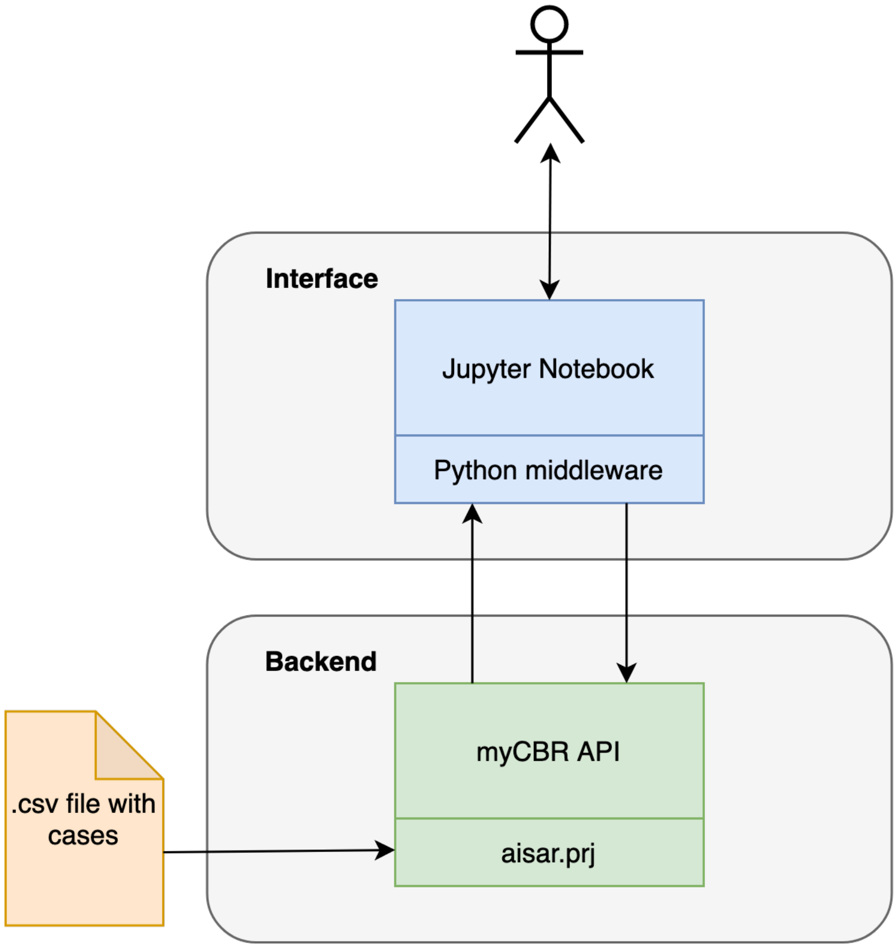

Figure 5.1: System overview

results of a measure using a data-driven approach to the *manual* measure. The *Rank* and *Score* measures have already been briefly introduced in Section 4.3.3 under global similarity measures. The paper uses Orange3 for access to a rank widget that provides several feature relevance scoring methods. Orange3 is an open source data mining tool that allows for visual programming [Orange3, 2020]. The visual programming tool contains widgets, which are building blocks for data analysis workflows. The widgets that have been used in the development of the algorithm can be seen in Figure 5.2.

The **file** widget imports a .csv file with cases, which is the same file that was illustrated in Figure 5.1. During import each attribute needs to be set to either *categorical*, *numeric*, *text* or *datetime*. The **data table** widget is used for a visual

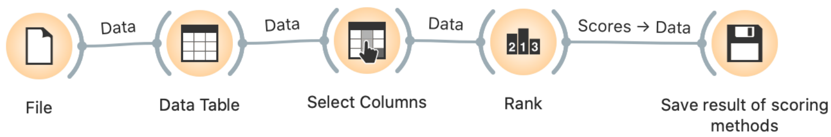

Figure 5.2: Workflow for finding scores given by various scoring methods

representation of the data in a spreadsheet and the data is sent to the **select columns** widget for manual composition of the data domain. Using manual composition of data it was possible to set the *action* attribute as the target attribute and the *result* and *incident id* attributes as meta attributes. This means that we want the feature relevance scoring methods to find the relevance of each attribute given the *action* attribute, while ignoring the *result* and *incident id* attributes.

The main widget in the workflow is **rank**, which uses the data and finds the score of each attribute using various feature relevance scoring methods. The feature relevance scoring methods used are *Information Gain, Gain Ratio, Gini, Chi2, ReliefF, FCBF* as described by Jaiswal and Bach [2019]. The scores are then saved to a .csv file using the widget called *save result of scoring methods*.

Algorithm 1 contains the pseudo code for the re-implemented, **Rank** algorithm, based on the paper by Jaiswal and Bach [2019]. Inputs for the algorithm includes the scores that were saved to the .csv file using workflow 5.2 and the percentage of the highest scoring attributes to be considered from each scoring method. The rank given to each attribute depends on the position of the attribute, which is sorted in descending order, and the number of attributes to be considered given by a percentage. The attribute that receives the highest score from a scoring method will receive a rank that is equal to the maximum number of attributes, *max_attributes* in Algorithm 1. If the current attribute has received the same score as the previous attribute, then the rank of the current attribute will be equal to the rank of the previous attribute. The number of attributes returned by the algorithm varies by the percentage, the scores of each scoring method and the number of scoring methods used.

The algorithm for **Score** is illustrated in Algorithm 2, and it is quite similar to Algorithm 1 for finding the rank measure. The difference in Algorithm 2 is that line 4-9 have been removed, as the weight for each attribute is based on the actual score instead of the rank received. In addition, we decided to make an alteration on the number of highest weighted attributes to be returned given by

Algorithm 1 Rank algorithm

Input: *scores* ← *a pandas data frame of all scores received from the scoring methods*

Input: *percentage* ← *percentage of attributes to be considered, between 0 and 1*

Output *weights* ← *dictionary of attribute, rank pairs*

```
function computeAttributeRanks(scores, percentage)
    max_attributes ← len(scores) * percentage
    for score_method_name, method_scores in scores do
        top_attributes ← method_scores.sort(descending).head(max_attributes)
        for attribute_name, score in top_attributes do
            if previous attribute exists and previous score equals score then
                rank ← rank of previous attribute
            else
                rank ← current index position
            attribute_ranks[attribute_name].append(rank)

    for attribute_name, rank in attribute_ranks do
        rank_sum[attribute_name] ← sum(rank)

    min_rank_sum ← min(rank_sum.values())
    max_rank_sum ← max(rank_sum.values())
    N = len(rank_sum)
    for attribute_name, rank_sum_per_attribute in rank_sum do
        weights[attribute_name] ← (N − 1) ∗ ((rank_sum_per_attribute − min_rank_sum)/(max_rank_sum − min_rank_sum)) + 1
return weights
```

the percentage. The percentage will choose the highest weighted attributes after each attribute has received a final global weight. As a result this selection is made on line 14 in Algorithm 2, so that only the percentage of the highest weighted attributes are returned. If a percentage of 50% is given to the Score algorithm and there are 32 attributes, then the number of attributes returned will always be 16.

Algorithm 2 Score algorithm

```
    Input: scores ← a pandas data frame of all scores received from the
    scoring methods
    Input: percentage ← percentage of attributes to be considered, between 0 and 1

    Output top_attributes ← dictionary of attribute, rank pairs
 1: function computeAttributeScores(scores, percentage)
 2:     max_attributes ← len(scores) * percentage
 3:     for score_method_name, method_scores in scores do
 4:         for attribute_name, score in method_scores do
 5:             attribute_ranks[attribute_name].append(score)
 6:     for attribute_name, rank in attribute_ranks do
 7:         rank_sum[attribute_name] ← sum(rank)
 8:     min_rank_sum ← min(rank_sum.values())
 9:     max_rank_sum ← max(rank_sum.values())
10:     N = len(rank_sum)
11:     for attribute_name, rank_sum_per_attribute in rank_sum do
12:         weights[attribute_name] ← (N − 1) ∗ ((rank_sum_per_attribute −
    min_rank_sum)/(max_rank_sum − min_rank_sum)) + 1
13:
14:     top_attributes ← weights.sort(descending).head(max_attributes)
15: return top_attributes
```

Figure 5.3 illustrates a flow chart of the Rank algorithm, in order to facilitate further understanding of the flow of the system. We have made some modifications in Figure 5.3, from the one presented by Jaiswal and Bach [2019]. Orange 3 returns the result of all of the feature relevance scoring methods, instead of one at a time. The result of each scoring method is then investigated by sorting the attributes in descending order by the score and selecting the percentage of highest scoring attributes that will be used. As mentioned earlier the rank is then given to each attribute. The formula for finding the weight of each attribute based on the rank is the same as the one presented in the paper by Jaiswal and Bach [2019]. It was decided to check if the Rank algorithm could be improved by utilizing other feature relevance scoring methods. Therefore, it was tested if the tool Weka [2020] with the feature relevance scoring methods *Symmetric Uncertainty, Information gain, Gain ratio, One R and ReliefF* would improve the Rank algorithm. However, there were no difference in results compared to using the feature relevance scoring methods that were included in Orange 3.

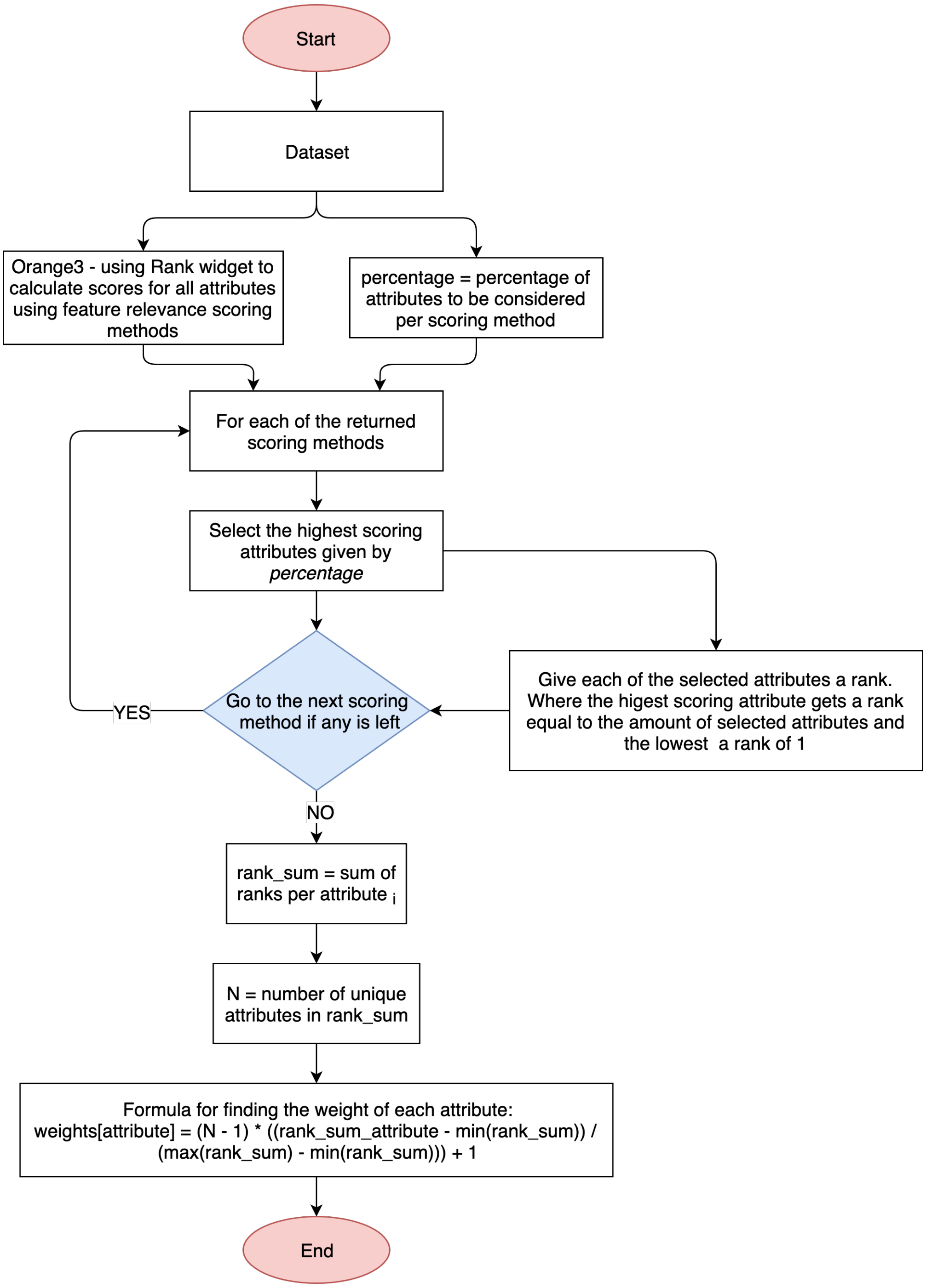

Figure 5.3: Flow chart of Rank algorithm as we have implemented based on the paper by Jaiswal and Bach [2019]

5.4 Endpoint for myCBR REST API

In order to easily evaluate the results of the Rank and Score similarity measures for different percentages of attributes an endpoint for the myCBR REST API needed to be created. The myCBR REST API did not have an endpoint for adding global similarity functions with weights. Most of the logic for this operation existed already in the myCBR SDK, so only small alterations were necessary to the code in the SDK before an endpoint could be made.

The endpoint was created for the myCBR REST API using Java and functionality needed to be added to the myCBR Python wrapper as well. The endpoint takes as input the *concept id*, *amalgamation id*, *amalgamation type*, and *weights* for each of the attributes using the json format. The *amalgamation id* is the name of the global similarity function that should be created, and the *amalgamation type* is weighted sum. The json format could look like $\{$"*weights*" : $[\{$"*goal*" : $40\}, \{$"*event*" : $30\}, \{$"*rain*" : $4\}]\}$, where the attributes *goal*, *event* and *rain* are existing attributes in the CBR component. The attributes that are not listed will receive a weight of 0 and the discriminant will be set to false. All new global similarity functions will use the local similarities created for each attribute.

5.5 Finding the Best Percentage of Weights

Algorithm 3 finds the best percentage of active attributes based on the sum of the MAP of the previous result. The algorithm uses the evaluation method Leave-one-out cross-validation (LOOCV) or k-fold cross-validation, given by the input attribute *loocv*. The functions *loocvCrossValidation* and *kFoldCrossValidation* perform the stated evaluations and will be discussed in Chapter 7. Line 13 of Algorithm 3 uses the endpoint we created for the myCBR REST API to create new global similarity measures. The main objective of the algorithm is to perform an evaluation of the optimal percentage of highest weighted attributes given the MAP for a given measure represented by the input attribute *type_measure*. In order to decrease run-time of the algorithm we used $percentage = 0.4$, so that the algorithm only checks for attributes equal to or less than 40%. The function *getPercentageWeights* takes as input a percentage that will be used to find the highest weighted attributes from a data set.

The endpoint makes it possible to automatically create several global similarity functions which varies in the number of activated attributes or the weights received. This will make it easy to test and find good global similarity functions.

Algorithm 3 Find good percentage of attributes

Input: *cbr* ← *an object of the myCBR REST API wrapper*
Input: *highest* ← *sum of Mean Average Precision for another similarity measure*
Input: *type_measure* ← *string of rank, manual, equal or score*
Input: *cases* ← *all cases in the case base, structure depends on loocv*
Input: *attr* ← *a sorted list of the possible values for the target variable*
Input: *loocv* ← *if True evaluation using leave-one-out cross-validation will be used, otherwise k-fold*

Output *cm* ← *confusion matrix of actual and predicted classifications*
Output *f1_scores* ← *a list of the f1-scores received*
Output *mean_avg* ← *a list of the MAP per k-round*

```
function findGoodPercentageAttributes(cbr, highest, type_measure, cases, attr, loocv = True)
    current ← 100
    percentage ← 0.4
    while current >= highest do
        if type_measure equals 'rank' then
            weights ← computeAttributeRanks(scores, percentage)
        else if type_measure equals 'manual' then
            weights ← getPercentageWeights(manual_weights,percentage)
        else if type_measure equals 'score' then
            weights ← computeAttributeScores(scores, percentage)
        else
            weights ← getPercentageWeights(equal_weights,percentage)
        cbr.createAmalgamationWithWeights(weights, name, type)
        if loocv then
            cm, f1_scores, mean_avg = loocvCrossValidation(cbr, cases, attr, name)
        else
            cm, f1_scores, mean_avg = kFoldCrossValidation(cbr, cases, attr, name)
        percentage− = 0.10
        current ← sum(mean_avg)
    return cm, f1_scores, mean_avg
```

Chapter 6

Case Example Run

This chapter will go through an example run of the system, so that it is easy to understand how the system works. First, Section 6.1 will look into an illustration of the system where each part is explained and discussed. In Section 6.2 a query case will be presented and used for retrieval in the CBR system. Furthermore, the similarity measures will be illustrated in Section 6.3 to show how retrieval is affected. Lastly, the retrieved, most similar, cases will be illustrated in Section 6.4.

6.1 System

In this section we will introduce and discuss Figure 6.1 and 6.2 for the example run of the system. All of the modeling discussed in Chapter 4 is contained in the aisar.prj file, illustrated in both figures. The aisar.prj file, which is explained in Section 5.1, contains the activated local similarity measure for each attribute, so this will not be part of the example run.

Figure 6.1 shows the workflow of Algorithm 3 for creating global similarity measures with a good percentage of activated attributes. The workflow shows how either *rank*, *score*, *manual* or *equal* are added to the aisar.prj file. The workflow uses Orange 3 and the rank widget to find the scores of the various scoring methods necessary for the *rank* and *score* measures, also illustrated in Figure 5.3. The next step is running Algorithm 3, where *type_measure* can have the values *rank, score, manual* or *equal*. If *rank* is chosen, then we will run Algorithm 1 and the weights returned by the algorithm are used by the myCBR REST API endpoint. During this example run we will use the global similarity measure, *manual_wt_30*, which we have created using the workflow illustrated in Figure 6.1. The function *findBestAvailableAttributes* will be given the input *type_measure* = *manual* and following the workflow we can see that the function *getPercentageWeights(i)* is used. The function *getPercentageWeights(i)* will read the manual weights stored in a .json file and find the percentage of attributes that have the best MAP. The result was the *manual_wt_30* measure, where 30% of the attributes were included.

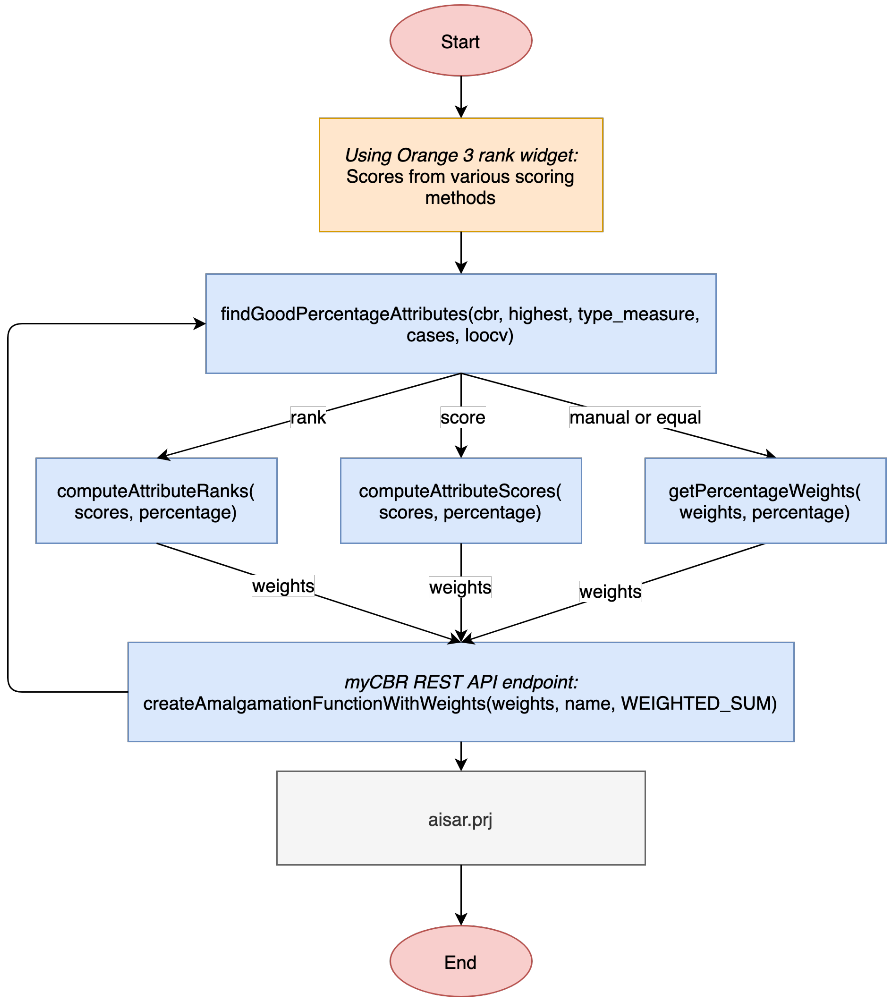

Figure 6.1: Workflow of Algorithm 3 on how to find the best global similarity measures

Looking at Figure 6.2 the global similarity measures that were created using the workflow in Figure 6.1 is now contained in the aisar.prj file. Figure 6.2 also illustrates Jupyter Notebook, which works as the user interface where the $problem = incident$ 12 has been chosen as the query case. In order to make a retrieval to the CBR component one of the names for an existing global similarity measure needs to be given to the endpoint. Here we have chosen to use the global similarity measure, *manual_wt_30*. An ephemeral case base is created and used for testing purposes. By creating an ephemeral case base, it is possible to use k-fold cross-validation, where the cases are split into k parts and only one part is retained in the ephemeral case base at any one time. However, we will use LOOCV with k=N where N is the number of cases. As will be explained in more detail in Section 7.1.

The myCBR REST API call *getSimilarCasesFromEphemeralCaseBaseWithContent* will create an instance of the ephemeral case base and make a query to the ephemeral case base to look for similar cases. The API call will then return a sorted list of all of the cases stored in the ephemeral case base, where an attribute stating the similarity of each case has been added. The rest of the chapter will go into detail and explain each of the components.

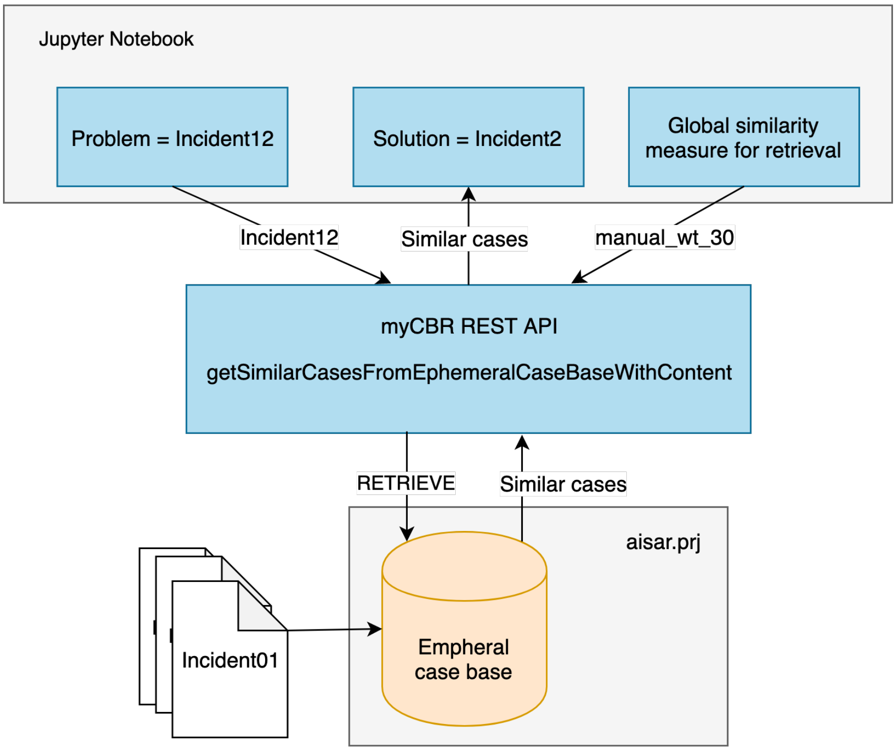

Figure 6.2: Illustration of system using Incident 12 as the query case

6.2 The Query Case

In order to show how the system works a query case is needed. The case chosen will be referred to as incident 12 and its attribute values are depicted in Table 6.1. These attributes make up the query case and will be used to make a retrieval to the CBR system. All of the attributes are known except *cloud_cover*, which is an unknown value and is written as _unknown_. The gray colored attributes *action* and *result* make up the solution context and will not be used in the query. The case content of incident 12 is based upon Chanko in Appendix A.1 and the concrete case can be found in Appendix A.2 with *incident_id = 12*. Incident 12 is based on a specific time frame during the accident where the vessel Chanko was involved. As the *goal* of the query case is to *limit_drift* the most similar case retrieved should recommend an action with the same sub-goal.

Table 6.1: Illustration of the query case, incident 12.

Attributes	**incident12**
category	grounding
event	drift
distress_call	panpan
emergency_phase	alert
pear	asset
daylight	FALSE
time_of_year	0.33
rain	0
temp_air	2
wind	26.7
cloud_cover	_unknown_
wave_height	10.0
conditions_worsen	TRUE
longitude	69.63
lattitute	17.82
offshore	TRUE
trafficated	FALSE
range_from_shore	short
radio_contact	FALSE
vessel_size	26.24
weight	145
ais	TRUE
building_year	1961
vessel_type	tug_boat
speed	4
pob	4
experience	some
training	some
mob_suit	TRUE

Table 6.1 continued from previous page

children_aboard	FALSE
nationality_crew	norwegian
goal	limit_drift
action	put_out_anchor
result	success

6.3 Similarity Measures

In order to enable retrieval of the most similar cases, local and global similarity measures will be used. Figure 6.3 shows some of the incidents with local similarity scores received for activated attributes. Which attributes are active is determined by the global similarity measure, *manual_wt_30*, also depicted in Figure 6.3. By using *manual_wt_30* only 30% of the highest weighted attributes are included in the measure. The highest weighted attributes represent the concepts *incident*, *environment*, *POB*, and *area* as illustrated in Ontology 4.1. Section 4.3.3 explained the difference between local and global similarity measures and some of the modeling choices made for the *manual* global similarity measure.

Figure 6.3 illustrates some of the incidents and the local similarity scores received by each attribute when compared to the query case. Looking at incident 2, we see that the incident has the same *goal* and *event* as the query case, incident 12, visible from the local similarity score of 1.0. Figure 6.3 also illustrates the global weights. It is reasonable to assume that incident 2 will be placed high on the list of similar cases, as *goal* and *event* have the highest global weight, with weights of 20 and 15 respectively. Looking at the local similarity for incident 14 and 19 it is safe to assume that these will be located lower on the list over similar cases, as *goal*, *PEAR* and *event* have a local similarity score closer to or equal to 0. This shows the importance of relative weighting of attributes in the global similarity measures, as incident 14 has a higher local similarity than incident 2 for all other attributes.

Table 6.2 shows a comparison of the attribute values for incident 2 and the query case, incident 12. The highlighted, purple rows are the attribute values used to calculate the similarity between the two cases restricted by the global similarity measure, *manual_wt_30*. So, the purple rows marked in the table are the same attributes that are also illustrated in Figure 6.3. The table confirm that the query case and incident 2 have the same *goal*, *event* and *PEAR*. We are now going to show the calculations for finding the similarity score of the query case and incident 2, which is based on Equation (4.2). In order to find the similarity score we will be using weighted sum. The following equation calculates the global similarity for

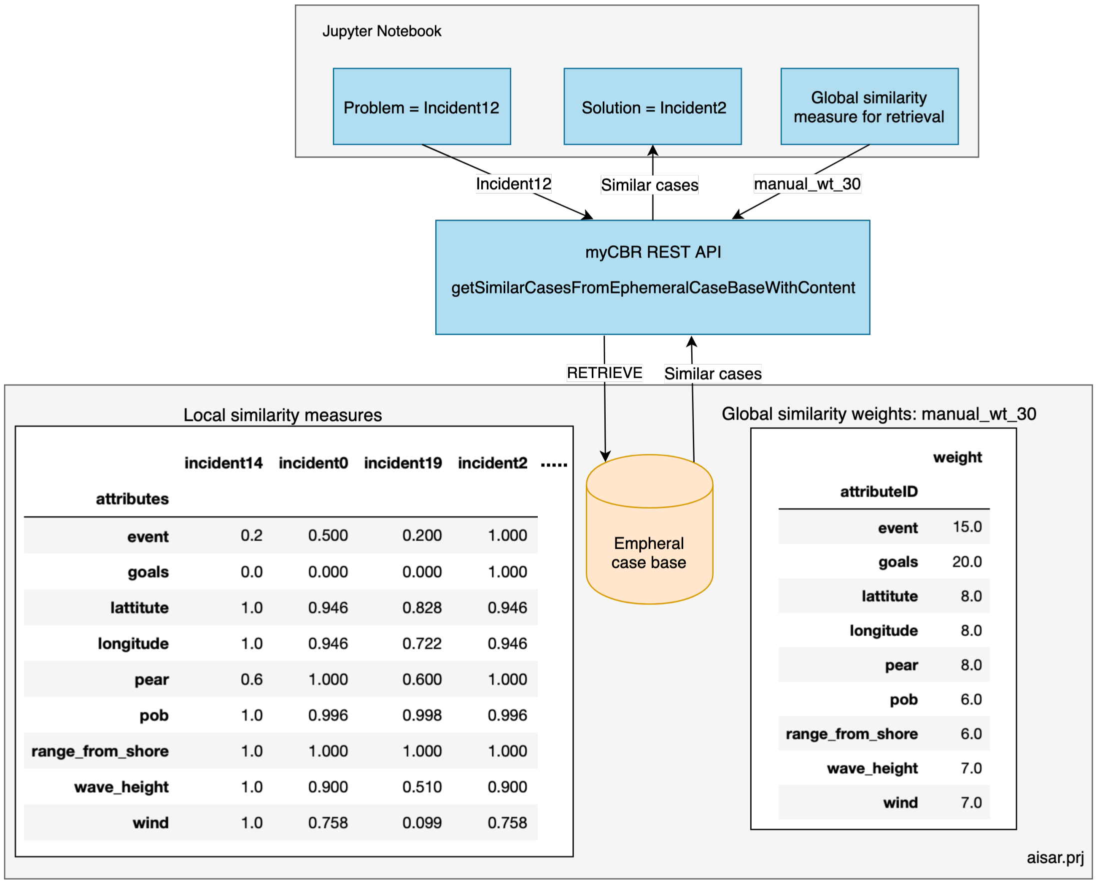

Figure 6.3: Illustration of system with similarity measures

each attribute where q is the query using Equation (4.1)

$$\begin{aligned}
Global\ similarity\ (q, incident2) &= \sum_i Global\ similarity_i(q, incident2) \\
&= 1.000 \cdot 15 + 1.000 \cdot 20 + 0.946 \cdot 8 \\
&\quad +0.946 \cdot 8 + 1.000 \cdot 8 + 0.996 \cdot 6 \\
&\quad +1.000 \cdot 6 + 0.900 \cdot 7 + 0.758 \cdot 7 \\
&= 81.718
\end{aligned} \tag{6.1}$$

In order to get a normalised similarity between 0 and 1, it is necessary to calculate the sum of all the global weights.

$$\begin{aligned} weights &= \sum_i global\ weight_i \\ &= 15.0 + 20.0 + 8.0 + 8.0 \\ &+8.0 + 6.0 + 6.0 + 7.0 + 7.0 = 85 \end{aligned} \tag{6.2}$$

Finally, the similarity score is calculated for the query and incident 2 using the result from Equation (6.1) and (6.2), giving us a normalised similarity score.

$$\begin{aligned} &Similarity\ score\ (q, Incident2) = \\ &\frac{1}{weights} \cdot Global\ similarity(q, incident2) \\ &= \frac{1}{85} \cdot 81.718 = 0.961 \end{aligned} \tag{6.3}$$

Table 6.2: Comparison of attribute values for incident 2 and the query case, incident 12.

Attributes	**incident2**	**incident12**
category	shipwreck	grounding
event	drift	drift
distress_call	mayday	panpan
emergency_phase	distress	alert
pear	asset	asset
daylight	FALSE	FALSE
time_of_year	0.08	0.33
rain	12	0
temp_air	3	2
wind	24	26.7
cloud_cover	TRUE	_unknown_
wave_height	8.0	10.0
conditions_worsen	TRUE	TRUE
longitude	67.63	69.63
lattitute	14.51	17.82
offshore	TRUE	TRUE
trafficated	FALSE	FALSE
range_from_shore	short	short
radio_contact	TRUE	FALSE

Table 6.2 continued from previous page

vessel_size	57.26	26.24
weight	969	145
ais	TRUE	TRUE
building_year	1952	1961
vessel_type	cargo_ship	tug_boat
speed	6	4
pob	6	4
experience	experienced	some
training	trained	some
mob_suit	TRUE	TRUE
children_aboard	FALSE	FALSE
nationality_crew	norwegian	norwegian
goal	limit_drift	limit_drift
action	put_out_anchor	put_out_anchor
result	success	success

6.4 Retrieved Similar Cases

Table 6.3 is a list of similar cases returned by the API call to the ephemeral database in Figure 6.2. The list is sorted by the similarity score in descending order. For easier discussions the sub-goals of each incident have been added to the table.

The solution context for the highest-ranking retrieved case is considered correctly classified if the action has the same sub-goal as the query case. The sub-goal for the solution context is retrieved by traversing the tree-structure illustrated in Figure 6.4. The figure has been created to represent the relationship of possible values for the goal, sub-goal and action attributes. Figure 6.4 illustrates the tree-structure using the goals, sub-goals and actions currently stored in the case base. However, the backend code holds the whole tree-structure represented in Appendix A.3. The reason that we are comparing sub-goals, instead of actions, is due to the small case base where an action might only be stored once. If the query case is the only instance with a given action in the case base, then the CBR system is unable to retrieve any cases with the same action. Often during a SAR incident all actions belonging to a sub-goal should be tried, so it was deemed reasonable that a retrieved action only needs to belong to the same sub-goal as the query case. For example, if the solution to the query problem is to *put_out_anchor* and the highest ranking retrieved case has the action, *start_main_engine*, then the classification will be considered correct as both have the sub-goal to *limit_drift*.

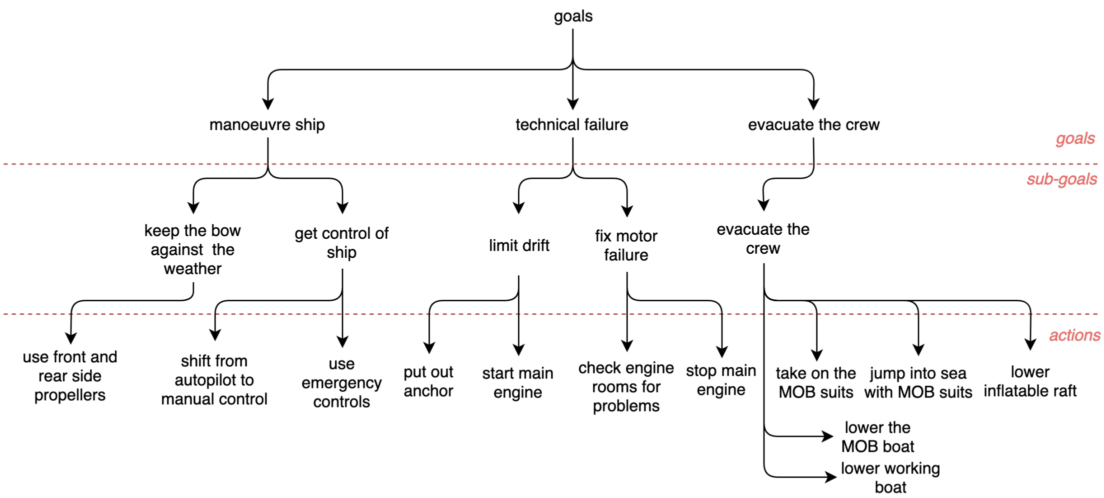

Figure 6.4: Tree structure over the goals, sub-goals and actions hierarchy

Incident 2, which was presented in Table 6.2, is at the top of the list over the retrieved cases in Table 6.3. All of the top three cases have the same sub-goal as the query case, incident 12. Incident 6 and 7 also have the sub-goal to *limit_drift*, but have a lower similarity score than incident 10 and 11, which have the sub-goal to *fix_motor_failure*. Looking at Figure 6.4 we can see that *fix_motor_failure* and *limit_drift* have the same main *goal* of fixing a *technical failure*. So, these results will have been affected by the taxonomy presented in Figure 4.9. All of the similarity scores in Table 6.3 have been found using Equation (6.3).

Table 6.3: Retrieved similar cases for incident 12 using similarity measure manual_wt_30.

Attributes	Similarity score	Sub-goal
incident2	0.961	limit_drift
incident8	0.924	limit_drift
incident9	0.924	limit_drift
incident11	0.835	fix_motor_failure
incident10	0.835	fix_motor_failure
incident7	0.818	limit_drift
incident6	0.818	limit_drift
incident5	0.691	fix_motor_failure
incident0	0.638	get_control_of_ship
incident1	0.638	get_control_of_ship
incident15	0.586	evacuate_the_crew

Table 6.3 continued from previous page

incident13	0.586	evacuate_the_crew
incident14	0.586	evacuate_the_crew
incident3	0.547	evacuate_the_crew
incident4	0.547	evacuate_the_crew
incident17	0.519	get_control_of_ship
incident16	0.519	get_control_of_ship
incident19	0.429	evacuate_the_crew
incident18	0.429	evacuate_the_crew

The user would be presented with the solution context of the case with the highest similarity score. Given the list illustrated in Table 6.3 the user would be presented with incident 2 as the best matching case. Table 6.4 illustrates the solution context, holding the *action* and *result* attributes, that would be illustrated for the user. The *goal* attribute has also been included to illustrate that it is in fact a matching case. The *action* and *result* attributes are the same as the those for the query, as illustrated in Table 6.3.

Table 6.4: Solution context of incident 2 presented to user as the most similar case.

Attributes	**Incident 2**
goal	limit_drift
action	put_out_anchor
result	success

Chapter 7

Evaluation and Results

This chapter will first introduce the evaluation methods and measures that will be used to evaluate the usefulness/quality of the system in Section 7.1. Section 7.2 will present the results of the evaluated prototype using two different evaluation methods. Additionally, we will look further into the relationship between local and global similarity measures for manual_wt_all. Lastly, a summarized interpretation of the results will be presented in Section 7.3.

7.1 Evaluation of the Global Similarity Measures

This section will address Research Question 6 presented in Section 1.3, "*How to evaluate the usefulness/quality of the system?*". It was decided that the global similarity measures would have the biggest impact on the quality of the implemented prototype, as the objective was to create a CBR component for the retrieval process. A comparison will also be made on the results of the global similarity measures for different percentages of activated attributes.

The evaluation of the global similarity measures has been re-implemented based on the same approach as described in the paper by Jaiswal and Bach [2019]. A confusion matrix will be created to illustrate the classifications results and we will use F1-scores as an evaluation measure. In order to gain a solid understanding of the retrieval results and ensure correct calculations the measures precision, recall, F1-score, average precision and MAP have been implemented without the use of libraries. The precision, recall and F1-scores are calculated for the solution context of the highest ranking case returned by a query to an ephemeral case base. The F1-score that was implemented is illustrated in equation (7.1) and the measure is used to create a box plot for a visual comparison of the global similarity measures. The F1-scores calculates the harmonic mean of precision and recall, which is often used for unbalanced data sets.

$$F1 - score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{7.1}$$

10-fold cross validation is the evaluation method used by Jaiswal and Bach [2019], however it soon became evident that the SAR case base was too small for this approach. As a result, 3-fold cross validation and LOOCV were implemented to evaluate the quality of the retrievals. Both evaluation methods are used as it allows us to compare how the results vary with the amount of cases in the case base. K-fold cross-validation randomizes the data set and then split it into K parts, where we decided to use K=3. It was decided to also evaluate the system using a MAP graph for 3-fold cross-validation. The average precision is calculated for the top three cases retrieved from an ephemeral case-base containing only one of the split data sets. An assumption had to be made for the calculation of the average precision as the data set is small and there is a randomization when the data set is split into three parts. So, if there are no relevant cases with the same sub-goal in the data set, then the average precision is set to 1.

LOOCV is similar to K-fold cross validation except that K is equal to N, where N is the number of cases in the case base. The MAP for a retrieval using LOOCV is therefore equal to the average precision score for the top three results retrieved by a query case.

The similarity comparison has been developed in Jupyter Notebook, which accesses the myCBR REST API running on the aisar.prj file created through the myCBR Workbench.

7.2 Results

In order to give better insight into the results of the global similarity measures, the results from both 3-fold cross-validation and LOOCV will be presented. We will also look into the local similarity scores and the global similarity measure, *manual_wt_all*, for incidents with the goal *limit_drift*.

7.2.1 3-fold cross-validation

Figure 7.1 shows confusion matrices of the classification results for the four different global similarity measures. Each column holds one of the four global similarity measures that were described in Section 4.3.3 and will be compared for different percentages of activated attributes. Figure 7.1 consists of three rows, where the first includes all of the attributes, the second row has a percentage of 50% and the last row of results has a varying percentage. The percentage of the last row depends on the result of Algorithm 3 introduced in Section 5.5. The y-axis of Figure 7.1 contains the actual values and the x-axis contains the predicted values, so correct classifications should be aligned on the diagonal axis.

Examining Figure 7.1 for 3-fold cross-validation, the *manual* measure we created based on domain knowledge acquired through literature research and the use of the *score* measure have better classification results for the first row. *Score_wt_50* in the second row predicts wrongly in one additional instance compared to *manual_wt_50* in the same row. Looking at the *rank* measure, we can see that for the first two rows the number of incorrect classifications is high compared to those of the *manual* and *score*. *Rank_wt_all* could be affected by the fact that the percentage of attributes chosen for *rank* is different compared to *score* and *manual*. Considering this, the *rank* measure will have a high likelihood of containing more attributes, as described in Section 5.3.

When looking at the last row, the measures *manual*, *rank* and *score* have the same number of correct classifications. Additionally, the figure shows that the same instances are wrongly classified, where e.g. *fix_motor_failure* was predicted when the actual label should have been *limit_drift*. In Section 5.3 for local similarity measures Figure 4.9 was introduced, illustrating the local similarity function for *goal*. This figure shows that *limit_drift* and *fix_motor_failure* have the same node as a parent and should have a higher local similarity than e.g. *limit_drift* and *evacuate_the_crew*. Since, 3-fold cross validation was used as the evaluation method it is possible that the data set did not contain any other instances of *limit_drift*. The *equal* measure performs worse than all of the other measures, so it becomes evident that the weights of the attributes are important to capture the necessary knowledge.

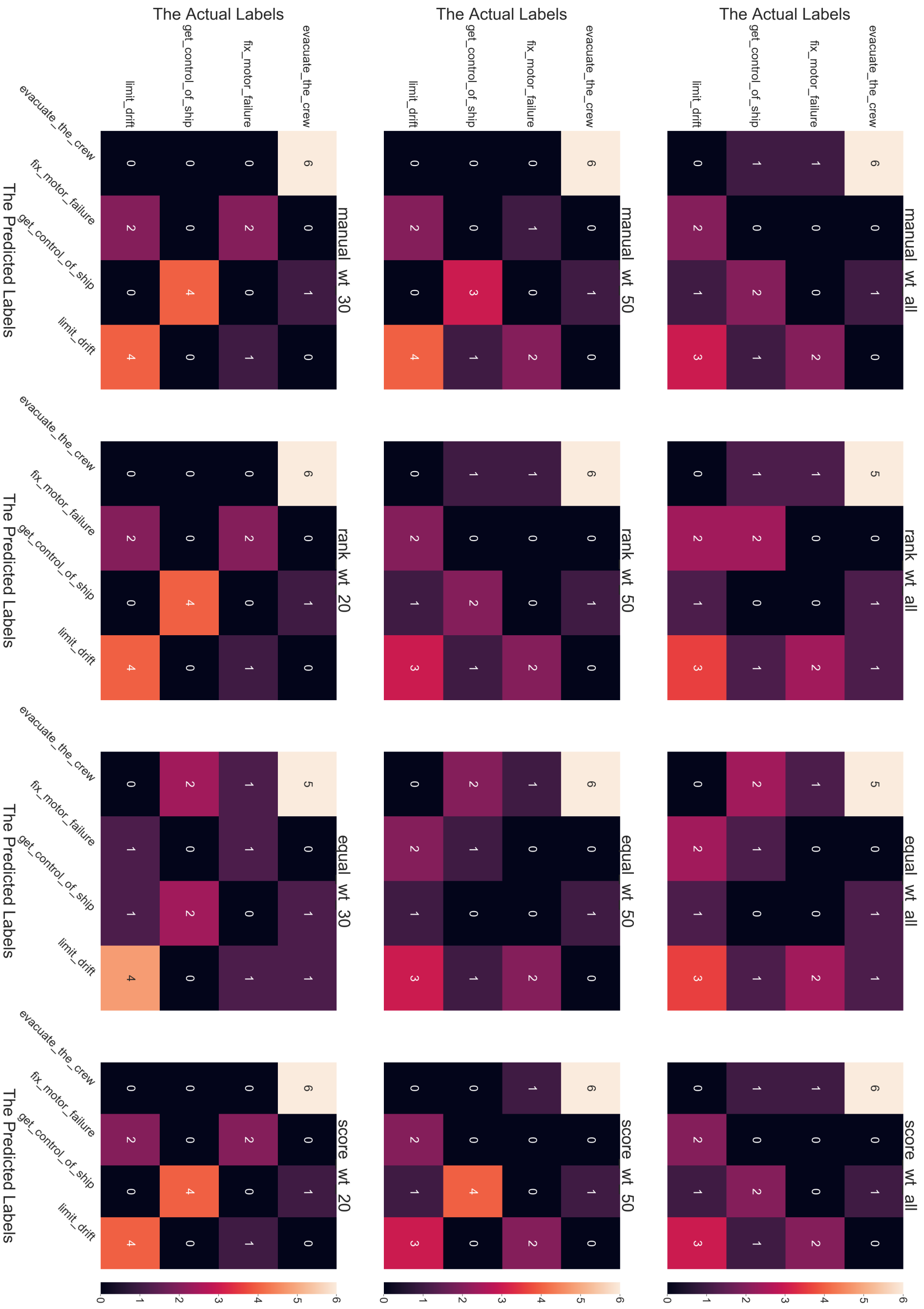

Figure 7.1: Confusion matrix of the different global similarity functions for 3-fold cross-validation

The F1-score is calculated for each confusion matrix generated from the results of retrieval at K=1, K=2 and K=3. The F1-scores have been plotted in Figure 7.2 where all of the global similarity measures have been included. The box plot shows the same trends as were discussed for Figure 7.1. The *equal* measure performs worse for all attribute percentages with the lowest F1-scores. As stated by Jaiswal and Bach [2019] it is possible to check if all or only a subset of attributes is necessary for a classification task, where the subset of attributes are chosen based on a percentage that choose the attributes with the highest global weight. The fact that all measures perform better when a lower percentage of highest weighted attributes are activated compared to when all attributes are included, could mean that some attributes might be redundant for the classification of actions. The *rank* and *score* measures were important to include in the box plot, as the captured domain knowledge should be able to have an equal or better performance than measures based on a data-driven approach.

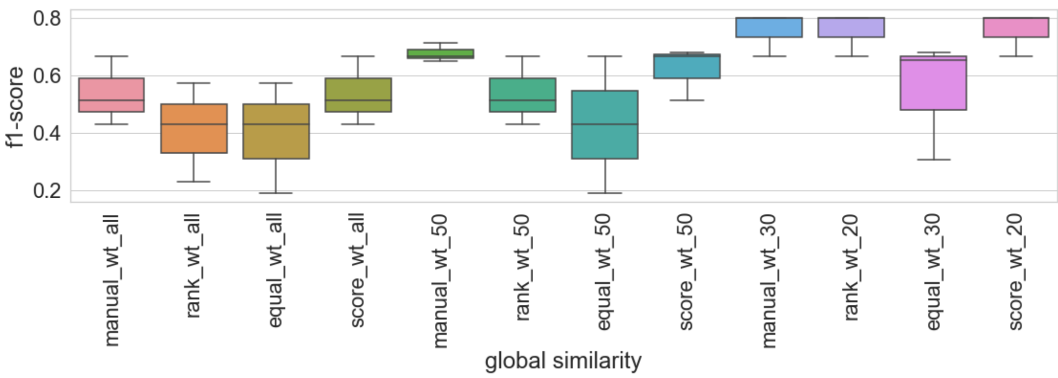

Figure 7.2: Box plot of the different global similarity functions for 3-fold cross-validation

The confusion matrices and box plot for 3-fold cross-validation will vary with each run, as the data set is randomized before it is split into K=3 parts. However, after several runs of the 3-fold cross-validation there is always the same trend that all of the measures, except *equal*, performs better with less attributes.

Figure 7.3 shows that all of the measures have a better MAP than the measure with *equal* weights. The measures that were plotted are the last row of Figure 7.1. The MAP is calculated for the top three retrieved cases, given a query case. The query is made to an ephemeral case base containing only one out of K-parts of the data set, where K=3. These results show that when we only consider the top three retrieved cases all of the measures, except *equal*, have a MAP of 1.0. As

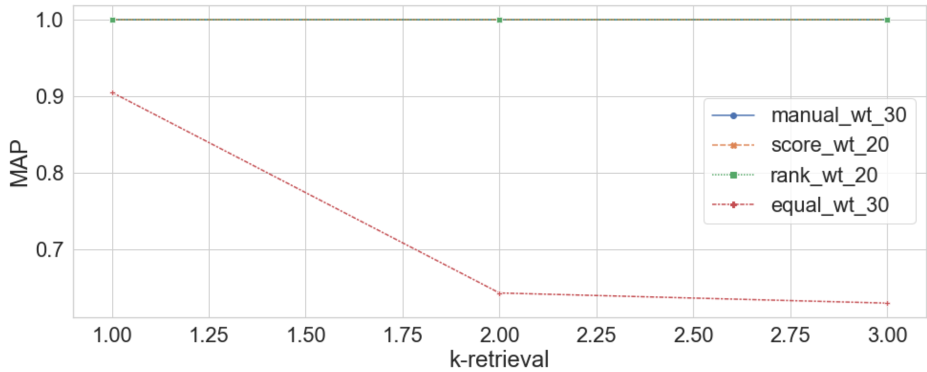

Figure 7.3: MAP graph for 3-fold cross-validation. The *manual*, *score* and *rank* measures all have a MAP of 1.0.

previously stated, an assumption was made due to the small data set when K=3, that if there are no relevant cases in the data set, then the average precision is set to 1. This is one of the reasons for the high MAP score in Figure 7.3.

7.2.2 Leave-one-out cross-validation

Figure 7.4 contains the confusion matrices of the results from the evaluation method LOOCV. LOOCV has more correct classifications than 3-fold cross-validation for all of the global similarity measures on the first and second row of Figure 7.4. When using LOOCV every query is made to an ephemeral case base containing all cases and not only a part of it. For all of the measures, except *equal*, there are no incorrect classifications on the last row. Looking at the figure we can see that the *equal* measure performs worse for the last row where 40% of the attributes are activated. It is evident that using Algorithm 3 that starts with a percentage of 40% for the *equal* measure yields insufficient results. The percentage of attributes are chosen at random for the *equal* measure, as there are no attributes with a higher weight to prioritize. The *manual*, *score* and *rank* measures perform better when a reduced number of attributes are included. The attributes that are included varies for each of the measures depending on which of the attributes had the highest weight. It is interesting to note that *manual_wt_50* and *score_wt_50* in the second row have equally good results and are both making one wrong classification for the same actual and predicted value.

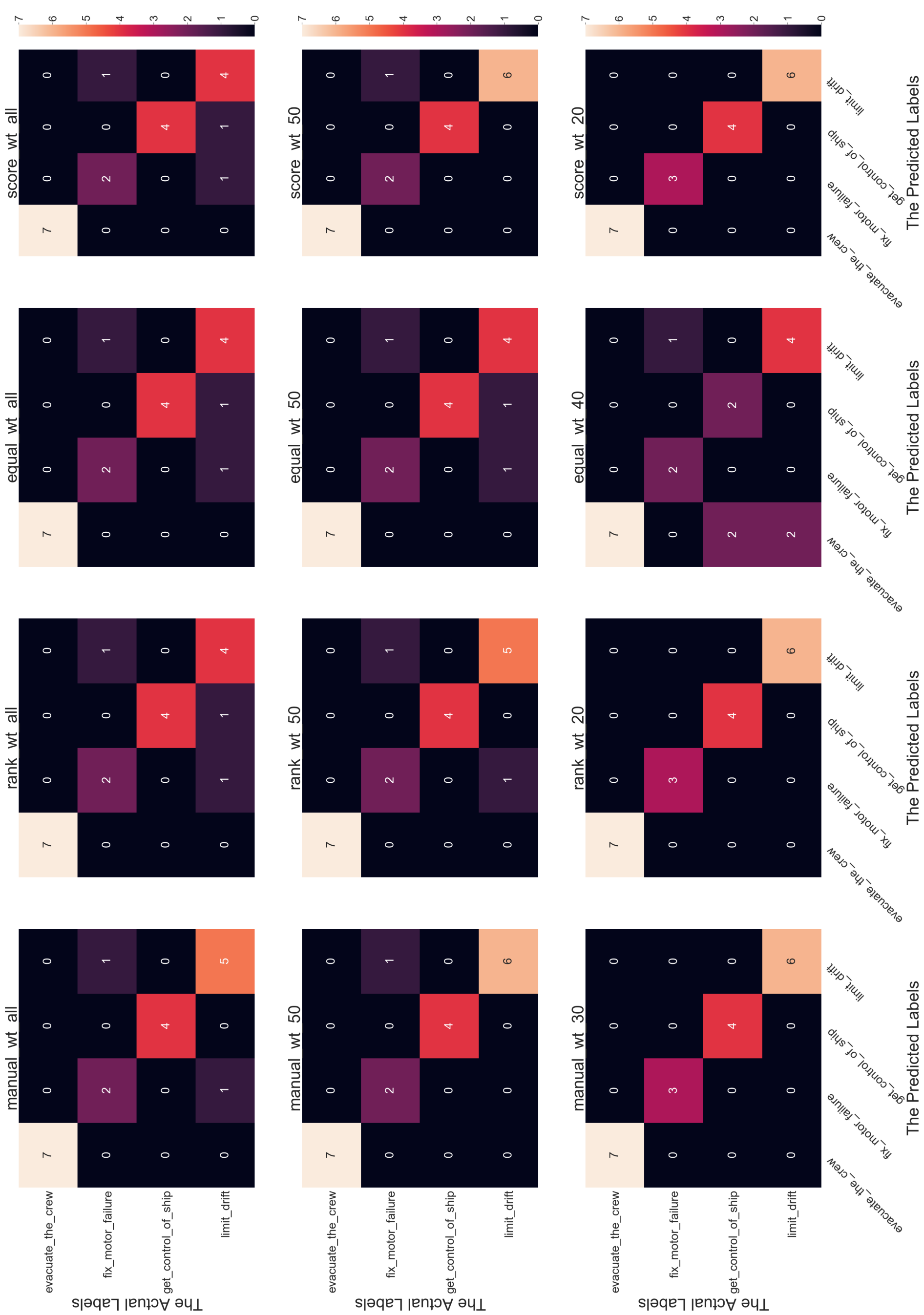

Figure 7.4: Results of the different global similarity functions for leave-one-out cross-validation

The box plots in Figure 7.5 have a larger number of F1-scores for each of the measures, as LOOCV is the same as k-fold cross-validation, except that k is equal to N. Where N is the amount of cases, so there will be 20 F1-scores for each measure. The *manual* measure shows better results on the first row, than all the other measures. On the second row both *manual* and *score* perform better than *rank* and *equal*. This can also be seen in the confusion matrices in Figure 7.4. The F1-scores capture a measure's accuracy given the case base, which is why the results for both 3-fold cross-validation and LOOCV have been included.

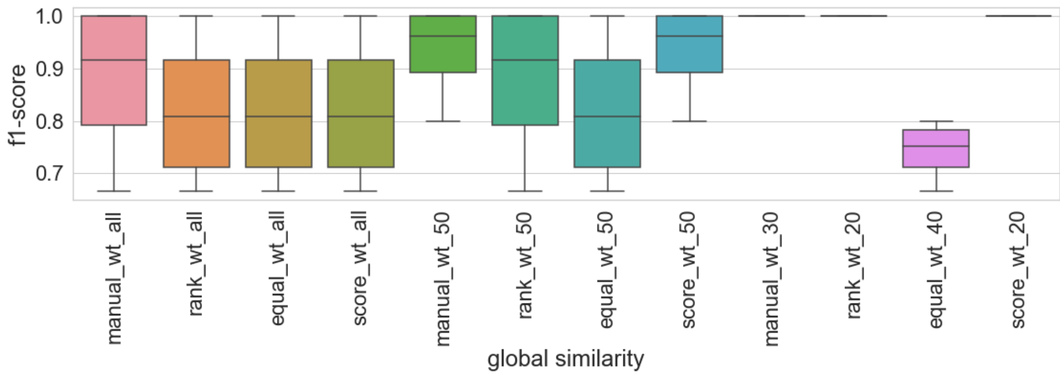

Figure 7.5: Box plot of the different global similarity functions for leave-one-out cross-validation

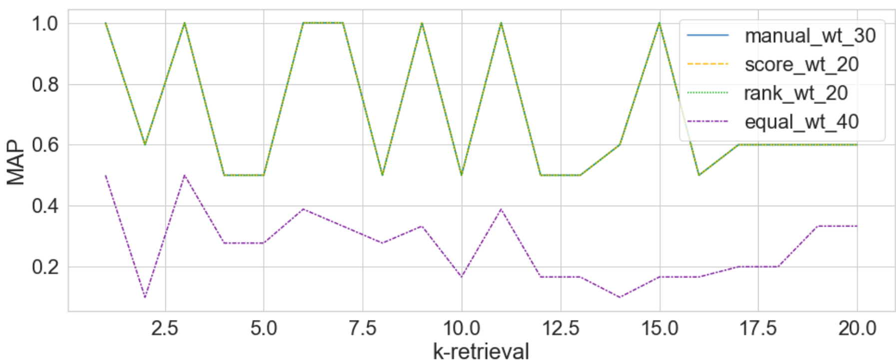

Figure 7.6: MAP graph for LOOCV. The *manual*, *score* and *rank* measures are overlapping in the figure.

The LOOCV results stay the same unless the case base is changed either by deleting or adding cases. Another way of displaying the results are by using a MAP graph. See Figure 7.6, which illustrates the MAP for the last row of measures in Figure 7.4. The average precision is calculated for the top 3 results retrieved for each query. In LOOCV the MAP will be the same as the average precision for each query. The graph plots the MAP for each of the retrievals $K = 1, 2, 3..., 20$. Figure 7.6 illustrates that all of the measures have a better MAP than *equal*. However, the graph also shows that there are often cases containing different sub-goals than the query in the top three retrieved cases.

7.2.3 Results of manual_wt_all for limit_drift

We will now discuss a self-similarity matrix and illustrate a heat map over the local and global similarity measures. In order to further illustrate the importance of the relationship between global and local similarity measures.

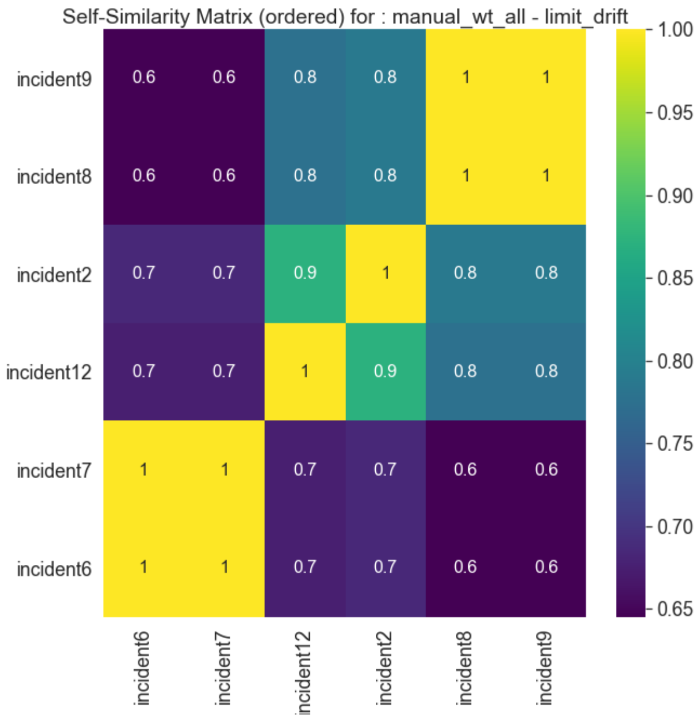

Figure 7.7: Self-Similarity matrix for cases with the goal *limit_drift*

Figure 7.7 displays a self-similarity matrix for incidents with the goal *limit_drift*. For incidents found in the AIBN reports multiple actions were tested for each goal.

So, some cases are stored with multiple instances in the case base where only the solution context containing the *action* and *result* attributes have been changed. The self-similarity matrix has used the *manual_wt_all* measure to retrieve the cases. Since the *action* and *result* attributes are part of the solution context, their global weight is set to 0. It is evident that this is the case in Figure 7.7 when observing the self-similarity matrix as incident 6 and 7 both have a similarity of 1. The same applies for incident 9 and 8. The *manual* measure contains all of the attributes, so that it is possible to discuss results where all of the attributes from the problem context are included. *Manual_wt_all* measure will be used for figures 7.8 and 7.9 and each incident will have the same similarity score as pictured in Figure 7.7.

For every attribute type, except Boolean attributes, local similarity functions were created through domain knowledge we had acquired. The similarity of cases where the goal is *limit_drift*, with incident 12 as a query case, can be seen in Figure 7.8. Incident 12 has been introduced as an example in Chapter 6 and its attribute values were illustrated in Table 6.1. Incident 12 is also represented in Figure 7.8, but all attributes have a local similarity of 1 as it is an exact match to itself. Figure 7.8 allows one to look at the local similarity between attribute values before the global weight is added to each attribute. Incident 2 is the most similar case to incident 12 and some of the attributes belonging to incident 2 have a lower local similarity than for some of the other cases. We can also see that incident 7 and 6 receive low similarity scores by looking at Figure 7.7. This is because *manual_wt_all* uses all attributes, as can be seen in Figure 7.8, and many of the attributes have a low local similarity score even though the sub-goal is the same.

Figure 7.9 shows the global similarity of each attribute with incident 12 as the query case, which is compared to all incidents with *limit_drift* as the goal. The global similarity for each attribute is normalized to have a value between 0 and 1 in order for the weighted sum to have a maximum of 1. The color map is scaled from 0 to 0.14, so it is easier to discern differences in attribute values, as the highest weight for an attribute in this figure is 0.14. In Figure 7.9 we can see the trend that all incidents have the same *goal* and *event*, but that the *event* attribute has a lower global similarity weight than *goal*. This figure also highlights the difference between incident 12 and incidents 6 and 7.

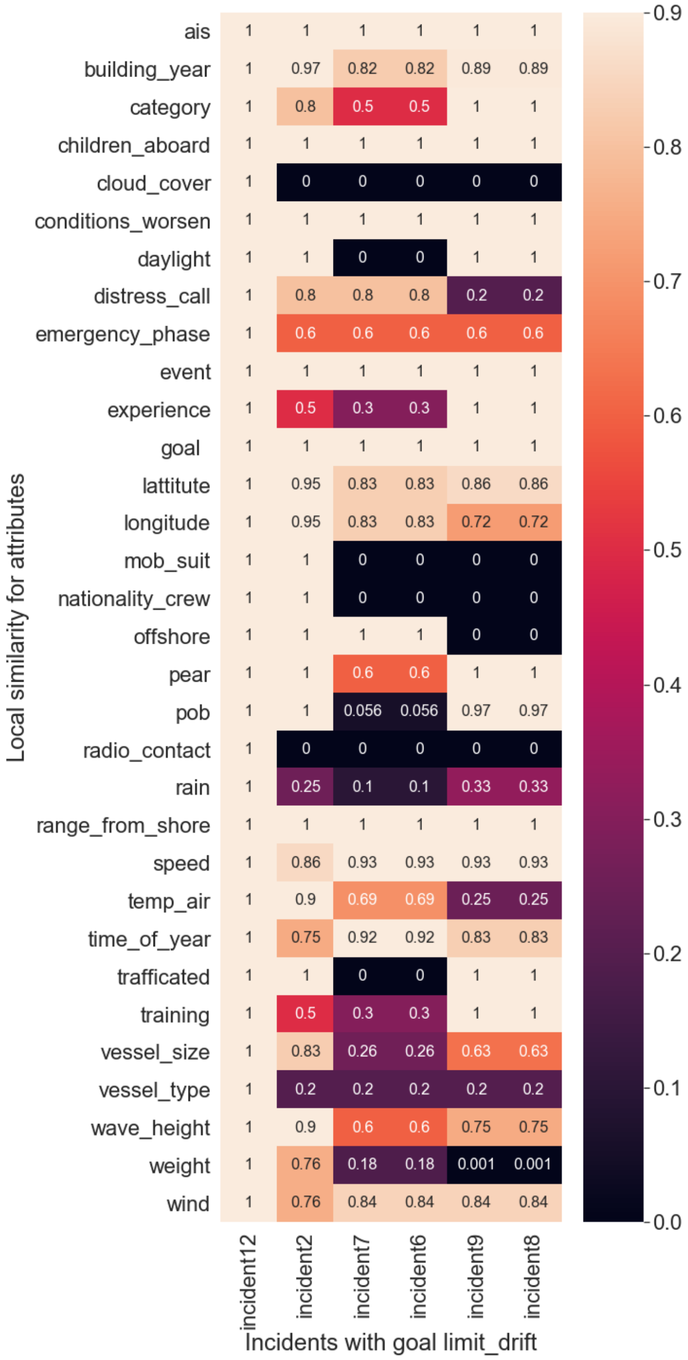

Figure 7.8: Local similarity for cases compared to incident 12 with the goal *limit_drift* using *manual_wt_all*

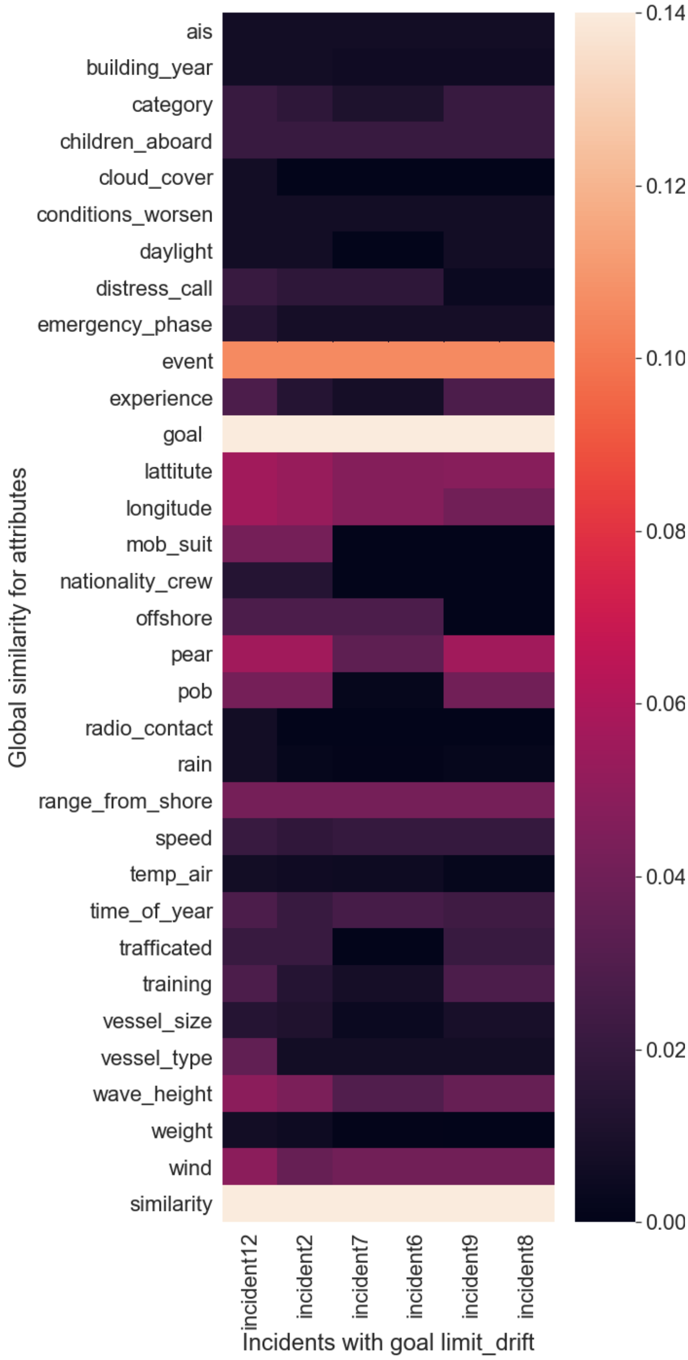

Figure 7.9: Global similarity for cases compared to incident 12 with the goal *limit_drift* and similarity measure *manual_wt_all*

7.3 Interpretation of Results

This section will give a summarized interpretation of the results. The section will present a summary of the results for the global similarity measures and the percentage of attributes.

7.3.1 Comparison of global similarity measures

The global similarity measures have been compared using confusion matrices, box plots of F1-scores and a MAP graph. Evaluations were performed using both 3-fold cross-validation and LOOCV. It should be repeated that the evaluation methods are comparing the goal of the query problem with the sub-goal of the retrieved case using a tree-structure for finding the sub-goal, given the action, see Figure 6.4 and Section 6.4 for an explanation. Both of the evaluation methods agreed on the fact that less attributes yielded better classifications results and F1-scores.

If we compare the last row of the confusion matrices in Figure 7.1 and 7.4 we can see that the number of correct classifications differ in four instances. The fact that 3-fold cross-validation gives four additional incorrect classifications compared to the results given by LOOCV, implies that the size of the case base affects the results. It was decided to use both evaluation methods based upon the findings in the related work for Section 3.3, where both papers found that results were affected by the knowledge and size of the case base. When using 3-fold cross-validation an ephemeral case base was created containing approximately 6 random cases, as the whole data set consisting of 20 cases must be divided into 3 separate sets. It is therefore safe to assume that the small data set is causing some of the incorrect classifications for 3-fold cross-validation. As the query case might be making a retrieval from an ephemeral case-base containing no previous cases with the same sub-goal or action. This is further confirmed by the fact that evaluation using LOOCV has no incorrect classifications on the last row of Figure 7.4. Figure 7.3 containing the MAP graph for 3-fold cross-validation also confirms this point as the MAP is one for each k-retrieval on the x-axis. An average precision of 1 was given if no cases existed with the same action or sub-goal. These findings confirm that it is important for a CBR system to have enough cases stored in the case base for good retrieval.

Looking at Figure 7.2 and 7.5 containing the box plots from both evaluation methods, we could see that the *manual* measure performs as well or better than the other measures for each group of attribute percentages. As a result, we choose to argue that the domain knowledge captured in this measure has been successful.

7.3.2 Percentage of attributes

The evaluations of the global similarity measures for various percentages of highest weighted attributes yielded interesting results. The attributes that should be activated for the *manual* and *score* measures were determined by the percentage and what attributes had the highest global weight. In contrast to the *rank* measure where the percentage is applied to the highest scoring attributes for each of the feature relevance scoring methods. The results presented by the box plots and the confusion matrices for both 3-fold cross-validation and LOOCV have demonstrated that the *manual*, *score* and *rank* measures have a higher rate of correct classifications, and better F1-scores when a subset of the attributes are activated. This leads us to the hypothesis that only a subset of attributes is necessary to correctly classify the recommended action given a matching sub-goal. Table 7.1 illustrates the activated attributes for the measures on the last row of Figure 7.4, where all classification results were correct. We also decided to test what would happen if we activated the attributes that were inactive for *manual_wt_30*, but active for *rank_wt_20* or *score_wt_20*. Using LOOCV it was possible to achieve classification results that were all correct if we activated the attributes for *manual_wt_30* that were regarded important by *rank_wt_20* and *score_wt_20*. This meant that the total number of activated attributes for the *manual* measure could be increased from 9 to 15, while maintaining the same results as those given by *manual_wt_30*. It could be interesting to investigate this further in future work to check if the percentage of attributes should not only be decided by its weight, but by using a combination of attributes as given by other measures.

Table 7.1: Activated attributes for the measures with the best classification results as presented in Figure 7.4. Gray rows represent id or solution context attributes.

Attributes	**manual_wt_30**	**rank_wt_20**	**score_wt_20**
action	0.0	0.0	0.0
ais	0.0	0.0	0.0
building_year	0.0	0.0	0.0
category	0.0	0.0	0.0
children_aboard	0.0	0.0	0.0
cloud_cover	0.0	1.33	0.0
conditions_worsen	0.0	0.0	0.0
daylight	0.0	0.0	0.0
distress_call	0.0	0.0	0.0
emergency_phase	0.0	0.0	0.0
event	15.0	10.33	23.35
experience	0.0	0.0	0.0
goals	20.0	12.0	32.0

Table 7.1 continued from previous page

incident_id	0.0	0.0	0.0
lattitute	8.0	1.67	0.0
longitude	8.0	0.0	0.0
mob_suit	0.0	0.0	0.0
nationality_crew	0.0	0.0	0.0
offshore	0.0	0.0	0.0
pear	8.0	2.33	0.0
pob	6.0	1.33	0.0
radio_contact	0.0	1.67	0.0
rain	0.0	1.0	0.0
range_from_shore	6.0	0.0	0.0
result	0.0	0.0	0.0
speed	0.0	7.33	16.04
temp_air	0.0	0.0	0.0
time_of_year	0.0	0.0	0.0
trafficated	0.0	0.0	0.0
training	0.0	0.0	0.0
vessel_size	0.0	1.0	11.09
vessel_type	0.0	0.0	0.0
wave_height	7.0	0.0	0.0
weight	0.0	1.33	11.10
wind	7.0	1.67	10.9

Chapter 8

Discussion

The main objective of this chapter is to address the research questions in Section 1.3 by discussing the results and our findings. Section 8.1 will discuss the ontology that was created for the SAR domain. A case representation and cases were built upon the terminology extracted for the ontology and will be discussed in Section 8.2. Next, Section 8.3 will discuss the similarity measures that were used to enable good retrieval. Lastly, we will discuss how the system was evaluated in Section 8.4.

8.1 Ontology for SAR Domain

This section will address the following research questions

RQ1: *What terminology does HRS use in order to explain a situation at sea that can inform building an ontology?*

RQ2: *How can we use this terminology to build a case and an ontology?*

The terminology that HRS uses to explain a situation at sea can largely be found in the IAMSAR [2010] manual. The manual gives a detailed overview of how SAR services should proceed during an incident. The categorization of SAR stages and emergency phases were introduced as important for SAR services, as this would provide helpful guidelines for what should be the main focus of attention. Each emergency phase was associated to a checklist that mainly consisted of information gathering actions, which are illustrated in Figure 4.2 of the developed ontology. The terminology for the ontology was extracted in an iterative manner, as more information about a SAR incident was gained. Section 4.1 presented details regarding the extracted terminology and how the ontology was built.

The IAMSAR [2010] manual is extensive, but it was found that other sources might give a different outlook on important terms. Thus, we did research in order to find different sources providing information of incidents at sea. Terminology for building an ontology was found in the timeline for the Viking Sky incident, an interview with a former mariner, and in the AIBN reports over marine incidents. For development of the ontology it was decided to use a middle-out approach, since the approach was found to include both theoretical modeling and text analysis. The ontology was built and expanded during each iteration of the extracted terminology. The final ontology is found in Figure 4.1 and 4.2.

8.2 Case Representation of Cases Populated in the Case Base

This section will address the following research questions

RQ3: *How can a case be represented and what will its content be?*

RQ4: *What cases found through research using the case representation found in RQ1 will be used to populate the case base?*

A lot of effort has been made on creating a good case representation for representing a SAR incident at sea. The literature review into related work presented a system called ASISA in Section 3.2.2. This was the only decision support system available for the SAR domain that was using a CBR component. However, the papers addressing the ASISA system did not give details regarding how incidents involving aeroplanes were represented. Therefore, the case representation for representing incidents at sea needed to be based on the terminology that we extracted from the developed ontology. The problem context for the case representation would be based on the situation assessment and a goal in order to predict the recommended action. The Snap system introduced in Section 3.3.1 also used the situation assessment and a goal to represent the problem context, which lead to successful results.

It was decided that a flat structure would be sufficient for representing a SAR incident using attribute-value pairs. The tool myCBR Workbench would be used for modeling of the case representation and the attributes were modeled as either Boolean, Integer, Float or Symbol attributes. All of the attributes that were chosen to represent a case were extracted from the developed ontology. Only the attributes that were regarded as important were extracted and the choice behind each was discussed in Section 4.3.1. The chosen attributes are represented in Figure 4.5 and an example of the case content was given in Table 4.1. Again, we used the AIBN reports and mapped information of an incident onto the case representation in order to verify modeling choices. The content of the case representation was also based on the AIBN reports, as possible values for an attribute were identified using real incidents. However, only a small amount of incident reports were read and the allowed values for some of the Symbol attributes will need to be expanded in the future when more cases are added.

The cases were built from the information we extracted from the AIBN reports. Appendix A.1 illustrates the cases where the fields marked with orange color represent educated guesses made from key words like *bad weather*. Information regarding the emergency phase that HRS would categorize an incident into was not present in the AIBN reports, and educated guesses on the value for the emergency phase was made. It was decided to use single value attributes, so the cases in Appendix A.1 were split into several different cases. The cases that were used to populate the case base is illustrated in Appendix A.2, where only the cases with the goal to *evacuate_the_crew*, *limit_drift*, *get_control_of_ship* or *fix_motor_failure* were included, as these goals had at least 3 entries. The case base was populated with a total of 20 cases. A higher number of cases in the case base would have been preferred, but the manual work of locating relevant attributes in the AIBN reports were too time consuming to increase the number of cases. Ideally, there should have been at least 3 cases for each specific action. The populated case base contains 20 cases where there are 11 different actions, so if the case base was to contain three cases for each action the total size of the case base would have to increase to at least $11 \cdot 3 = 33$. Section 4.3.2 discuss the cases that were used to populate the case base in more detail. One of the flaws of using cases based on the AIBN reports is the fact that these incidents had a severe outcome and thus the actions were often unsuccessful. Ideally, the case base should also contain cases where the outcome was positive.

8.3 Similarity Measures for Retrieval

This section will address the following research question

RQ5: *What similarity measures will be suitable to the attributes in the case representation?*

The similarity measures were introduced in Section 4.3.3. The configuration of the local similarity functions and how these have been modeled for each attribute type was discussed. A local similarity function is the comparison between two attribute values. The default local similarity in myCBR between two Symbol attributes is 0 if the values are different and 1 if there is an exact match. For

Integers and Floats the similarity is 1 no matter the value of the attributes. Therefore, it was decided that creating custom local similarity functions were important and would improve retrieval. Two local similarity functions for the *event* symbol attribute, see Figure 4.8, and the *goal* symbol attribute, see Figure 4.9, were presented as examples on two different local similarity functions for the symbol attributes. These local similarity functions made it possible for two attributes to have different values, but still receive a similarity higher than zero if the values were slightly similar. The plan was to verify the local similarity functions in the planned meeting with HRS that was canceled. Regardless, the local similarity functions we created based on our gained knowledge were considered better than the default functions provided by myCBR.

A lot of effort has been given to the creation of global similarity measures. The measures that we created were *manual*, *rank*, *score* and *equal*, where each one uses weighted sum to calculate the similarity score between two cases. A paper by Jaiswal and Bach [2019] was presented in related work describing a data-driven approach for finding global similarity weights. It was decided to re-implement the algorithm proposed in the paper, as the results were promising, giving us the measures *rank* and *score*. The *rank* and *score* measures would make it possible to compare classification results to the *manual* measure that we had created based on our obtained domain knowledge. The findings were that the *manual* measure managed to capture domain knowledge, as the results were better or as well as those for *score* and *rank*. All of the measures were found to give better results than the *equal* measure, which worked as a baseline.

It was decided after reading the paper by Jaiswal and Bach [2019] that it would be useful to check if the retrieval results were better for different percentages of activated attributes. Therefore, we created Algorithm 3 for finding a good percentage of attributes that should be activated for all measures based on the MAP. The results showed that the amount of correct classifications increased when the percentage of attributes that were activated decreased. It can be argued that some of the attributes provided noise for retrieval results given the small amount of cases in the case base and that using all attributes reduced the performance of the CBR prototype.

8.4 Evaluation of the System

This section will address the following research question

RQ6: *How to evaluate the usefulness/quality of the system?*

The goal was to design and develop the CBR component with focus on the retrieval process. Therefore, it was decided to evaluate the usefulness/quality of retrieval based on a comparison of different configurations of weights for the global similarity measures. The global similarity measures are important for good retrieval results. Section 7.1 introduced the evaluation methods that we decided to use, which were based on the paper by Jaiswal and Bach [2019] presented in related work. In the paper 10-fold cross validation was used to evaluate the global similarity measures and calculated F1-scores. Due to the small size of the case base it was decided to use 3-fold cross-validation and LOOCV. By comparing the results from both evaluation methods it was possible to also look into if the size of the case base has any impact. In addition to comparing the global similarity measures for various percentages of activated attributes.

It was found that the measures *manual*, *rank* and *score* performed better than the *equal* measure and that the number of correct classifications increased when less attributes were activated. Using LOOCV as the evaluation method gave a clear indication, when looking at the box plot of the F1-scores in Figure 7.5, that the measures *manual*, *score* and *rank* gave best results when using the percentage given by Algorithm 3. The number of correct classifications decreased when using 3-fold cross-validation as the evaluation method compared to LOOCV. Comparison of the results produced by each evaluation method verified our assumption that fewer cases might reduce the performance of a CBR system. This observation substantiates the findings of the papers presented in related work in Section 3.3.1 and 3.3.2. The incorrect classifications in 3-fold cross-validation are most likely caused by the fact that the data set has been split into three parts and might not contain a case with the same action or sub-goal.

Evaluation of the system was made using each case in the case base as a query case. One drawback of this approach is the fact that the system was not evaluated using queries containing partial information. When HRS is first notified of a potential incident there might be missing information that will take time to acquire. To also evaluate retrieval results using queries with partial information was regarded out of scope due to time constraints.

Chapter 9

Conclusion

Section 9.1 will present the conclusion based on the results in Chapter 7 and the discussion in Chapter 8. Next, Section 9.2 will look into how the current CBR prototype can be expanded in future work. We will also discuss the whole system that was designed in Section 4.2 and how this is also part of future work.

9.1 Conclusion

Our research found that CBR was a suitable approach for decision support during SAR incidents, due to the reduced need for domain experts compared to other KBS. The results of the evaluation methods confirmed that CBR is indeed the correct approach, as the system produced good results on classification, given solid global similarity measures.

It was vital to create a common ground for mutual understanding of SAR at sea, in order to explain incidents with the relevant and correct terminology. Therefore, focus was made on creating an ontology, for the benefit of structured information. The ontology proved useful as all attributes for the case representation could be carefully chosen from the ontology. The AIBN reports that were localized through extensive search proved to be indispensable for work on the thesis. The reports allowed us to verify most of the terminology used in building our ontology.

In the very beginning of our study we realized how complex the incident problem solving was and therefore decided to focus on the retrieval process of the CBR component. As a consequence, data modeling of the CBR prototype was a focal point for this project. The case representation was chosen carefully and verified by mapping information from the AIBN reports onto the attributes. As soon as the case representation was finalized the cases for the case base were created based on information from the reports. The fact that the case base consists of cases from real incidents increased the validity of our results. The local similarity measures were created using our understanding of the SAR domain and the importance of similarity between attribute values. The local similarity measure for each attribute was supposed to be verified by operators at HRS, but due to reasons out of our control this was not possible. However, it was concluded that local similarity measures based on our gained domain knowledge would prove better than none.

Extensive research and effort were made on creating good global similarity measures. The prototype was evaluated for different configurations of the global similarity measures. The findings showed that the *rank*, *score* and *manual* measures all performed better when there were a smaller percentage of attributes activated. In addition, these measures had a significantly better MAP score for all k-retrievals, than the *equal* measure. We could also conclude that the results improved with the size of the case-base by comparing evaluations using 3-fold cross-validation and LOOCV. The *manual* measure performed as well or better than the other measures for all percentages presented in Chapter 7. This inclines one to draw the conclusion that it is possible to develop a solid prototype for the retrieval process without access to a domain expert. However, the global similarity measures should be evaluated by a domain expert to verify weighting choices.

9.2 Future Work

This section will discuss the future work of the project. First, we will discuss improvements that can be made on the current data model. Then, as this thesis has focused on the retrieval process of the CBR cycle, we will discuss future work on creating a complete CBR system. Lastly, we will discuss how this thesis is but a starting piece of a bigger project spanning the whole life span of an incident. In Section 4.2 we designed a complete system that addresses the CBR parts of what the bigger project could entail and the discussion will be based partly on this.

9.2.1 CBR retrieval process

For the data modeling of the CBR retrieval process a weakness is the fact that the meeting with HRS was canceled. As a consequence, modeling choices for the ontology, case representation and similarity measures could not be confirmed by domain experts. However, the results of the prototype we have developed are promising.

In future work it could be of high interest to increase the size of the data set over past incidents, as this has been the biggest limitation. The prototype we have implemented are automated given that the workflow for Orange 3 displayed in Figure 5.2 is run for a new/increased data set. Then it is possible to run the Jupyter Notebook which works as a user interface by following the user manual in Appendix B.1. The notebooks will automatically find the similarity weights using the data-driven approach and evaluate the measures for different percentages of attributes. All figures in Chapter 7 were generated automatically through the notebook. In addition, it will be possible to check if the *rank* and *score* measures give other results when based on a larger data set of incidents. The valid attribute values for the *goal* and *action* symbol attributes, among others, will need to be extended if a larger data set with additional values will be used for future work.

If a data set contains enough incidents with the same *action* it is possible to modify the evaluations to only regard a solution as correct if the exact same action is suggested. However, the AIBN reports lead to the belief that given enough time, all actions should be tested until a successful result is reached as no two incidents are ever the same. One limitation of basing all of the cases on the AIBN reports was that most cases had a severe outcome. Ideally, the case base should also contain cases where a vessel is in need of assistance, but the circumstances are not as dire.

Another aspect that is important to consider is the fact that evaluations on the system were performed using queries where the information was complete. According to the IAMSAR [2010] manual the information is usually partial as it takes time to gather complete information. As discussed in Section 8.4, the system should be evaluated using queries with partial information as part of future work to investigate how this can affect the results of the system.

Finally, we mentioned in Section7.3.2 that we evaluated the system by activating additional attributes for the *manual_wt_30*. These additional attributes were decided by which attributes were activated for either *rank_wt_20* or *score_wt_20*. The manual measure with additional attributes was evaluated using LOOCV and the results gave no incorrect classifications. Therefore, future work could investigate an approach based on a combination of a percentage of highest weighted attributes and activation of those recommended by the data-driven approaches.

9.2.2 CBR cycle

As we said earlier, we have focused on only implementing the **retrieval** process of the CBR cycle, due to time restrictions. Future work should expand the prototype created in this thesis to encompass the whole CBR cycle illustrated in Figure 2.2. Section 2.4.1 describes all of the processes in a CBR cycle as presented by Aamodt and Plaza [1994]. The next step is the **reuse** process that should enable the solution of the most similar case to be reused for the query problem. As it is unlikely that an incident at sea will be an exact match to a case in the case base, the solution of the most similar case needs to be adapted to be applicable to the query problem. This brings us to the **revise** process, which is needed to determine if the adapted solution is correct. If not, the solution needs to be revised. Here a decision needs to be made on whether a SAR operator at HRS will make the evaluation or if it should be tested in the real world. If the experience was considered useful it should be **retained** in the case base.

9.2.3 Complete system

HRS wants a decision support system spanning the whole lifespan of an incident. Therefore, a design of a complete system was created and illustrated in Figure 4.3. In this thesis we have focused on the retrieval process of the "Recommended action" in Part A of the figure. As discussed, the whole CBR cycle for "Recommended action" should be developed as part of the future work. Additionally, the "casual hypothesis" on the cause of an incident in Part A is important for SAR when the cause is unknown. A causal hypothesis is the foundation of any search planning according to the IAMSAR [2010] manual and should be updated whenever new information is received.

The CBR components in Part B and Part C are considered planning problems. Part B should create a plan depending on the situation assessment and whether search or rescue is necessary at the time. The plan should be executed in the environment in order to see if the solution needs to be revised. Part C will consider the situation assessment and the results of the previous plan in order to either modify the plan or the causal hypothesis if necessary. The situation assessment for all of the CBR components should include the attributes in the case representation we have created for Part A. However, more attributes for representing a plan will be needed for Part B and C. The prototype we have developed for finding global similarity measures can be utilized in future work for finding global similarity weights for the CBR components in the development of Part B and C. A complete system should also be expanded to encompass land and air incidents as well.

Bibliography

Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59.

Abi-Zeid, I. and Frost, J. R. (2005). Sarplan: A decision support system for canadian search and rescue operations. *European Journal of Operational Research*, 162(3):630–653.

Abi-Zeid, I. and Lamontagne, L. (2003). A functional decomposition model of case prosecution in joint rescue coordination centers and related projects. *TR 2000-150, R&D pour la défense Canada - Valcartier.*

Abi-Zeid, I., Lamontagne, L., and Yang, Q. (1999). Is cbr applicable to the coordination of search and rescue operations? a feasibility study. In *International Conference on Case-Based Reasoning*, pages 358–370. Springer.

Abi-Zeid, I., Nilo, O., Schvartz, S., and Morin, M. (2010). Towards a knowledge-based system prototype for aeronautical search and rescue operations. In *2010 13th International Conference on Information Fusion*, pages 1–8. IEEE.

AIBN (2020a). Marine. `https://www.aibn.no/Marine`. Last accessed on Feb 14, 2020.

AIBN (2020b). Published reports. `https://www.aibn.no/Sjofart/Published-reports`. Last accessed on Feb 14, 2020.

Baxter, K., Courage, C., and Caine, K. (2015). *Understanding your users: a practical guide to user research methods*. Morgan Kaufmann.

El Ghosh, M., Naja, H., Abdulrab, H., and Khalil, M. (2016). Towards a middle-out approach for building legal domain reference ontology. *International Journal of Knowledge Engineering*, 2(3):109–114.

Feret, M. P. and Glasgow, J. I. (1997). Combining case-based and model-based reasoning for the diagnosis of complex devices. *Applied Intelligence*, 7(1):57–78.

Hovedredningssentralen (2018a). Statestikk. https://www.hovedredningssentralen.no/wp-content/uploads/2019/01/HRS-Statistikk-2018.pdf. Last accessed on Oct 11, 2019.

Hovedredningssentralen (2018b). Timeline for viking sky. https://www.hovedredningssentralen.no/tidslinje-viking-sky/. Last accessed on Nov 07, 2019.

Hovedredningssentralen (2019). About. https://www.hovedredningssentralen.no/english/. Last accessed on Oct 11, 2019.

IAMSAR (2010). *IAMSAR volume II - Mission Co-ordination*. International Maritime Organization and International Civil Aviation Organization.

Jaiswal, A. and Bach, K. (2019). A data-driven approach for determining weights in global similarity functions. In *International Conference on Case-Based Reasoning*, pages 125–139. Springer.

Jupyter (2020). The jupyter notebook. https://jupyter.org. Last accessed on Feb 25, 2020.

Luger, G. F. (2008). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Pearson Education, 6 edition.

MarineTraffic (2020). Marine traffic. https://www.marinetraffic.com/no/ais/home/centerx:0.4/centery:62.8/zoom:4. Last accessed on Feb 19, 2020.

Markovits, H. (2012). *Model-Based Reasoning*, pages 2307–2310. Springer US, Boston, MA.

myCBR (2020). mycbr. http://www.mycbr-project.org. Last accessed on Mar 20, 2020.

Noy, N. F., McGuinness, D. L., et al. (2001). Ontology development 101: A guide to creating your first ontology. https://protege.stanford.edu/publications/ontology_development/ontology101.pdf. Last accessed on Jan 6, 2020.

Orange3 (2020). Data mining. https://orange.biolab.si. Last accessed on Feb 25, 2020.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77.

Raja, H. Z., Sormo, F., Vinther, M. L., et al. (2011). Case-based reasoning: predicting real-time drilling problems and improving drilling performance. In *SPE Middle East Oil and Gas Show and Conference*. Society of Petroleum Engineers.

Richter, M. M. and Weber, R. O. (2013). *Case-Based Reasoning: A textbook*. Springer.

Russell, S. J. and Norvig, P. (2010). *Artificial Intelligence A Modern Approach: Third Edition*. Pearson Education Limited.

Schvartz, S., Abi-Zeid, I., and Tourigny, N. (2007). Knowledge engineering for modelling reasoning in a diagnosis task: application to search and rescue. *Canadian Journal of Administrative Sciences/Revue Canadienne des Sciences de l'Administration*, 24(3):196–211.

Swain, M. (2013). *Knowledge-based System*, pages 1084–1086. Springer New York, New York, NY.

Ting, S.-P. and Zhou, S. (2008). Snap: A time critical decision-making framework for mout simulations. *Computer Animation and Virtual Worlds*, 19(3-4):505–514.

Weka (2020). Weka: The workbench for machine learning. `https://www.cs.waikato.ac.nz/ml/weka/`. Last accessed on May 6, 2020.

Yang, Q., Abi-Zeid, I., and Lamontagne, L. (1998). An agent system for intelligent situation assessment. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 466–474. Springer.

Yu, W. and Li, R. (2009). Maritime search and rescue ontology construction based on protege. In *2009 International Conference on Information Engineering and Computer Science*, pages 1–3. IEEE.

Appendices

Appendix A

Architecture/Model

A.1 Cases from AIBN Reports

This section shows the cases as they were extracted from the AIBN reports.

NOTES:	
	Orange color means that these fields are educated guesses
	Dark blue is just a visual aid to seperate enviromental data from boat data and so on

Attribute types	**Jøsenbuen**	**FFS Achilles**	**Astrid Sofie**
category	other_accident	shipwreck	shipwreck
event	fatality, drift	missing_vessel, grounding	fire
distress_call	overdue	none	mayday
emergency_phase	alert	none	distress
pear	person	person	person
daylight	TRUE	FALSE	TRUE
time_of_year	0.83	0.25	0.25
words describing the weather:	*bad weather*	*good weather*	
rain	10	0	0
wind	15	8.2	25.9
cloud_cover	TRUE	FALSE	TRUE
wave_height	1	0.2	4
conditions_worsen	TRUE	FALSE	TRUE
longitude	59.15	58.08	59.47
lattitute	5.41	6.82	5.05
offshore	FALSE	FALSE	TRUE
trafficated	TRUE	FALSE	FALSE
range_from_shore	short	short	medium
radio_contact	FALSE	FALSE	TRUE
vessel_size	8.7	30.21	32
weight	100	285	430
ais	FALSE	TRUE	TRUE
building_year	2015	1984	2016
vessel_type	fishing_vessel	tug_boat	fishing_vessel
speed	5.4	8.4	1.7
pob	2	3	6
experience	some	some	experienced
training	some	some	some
mob_suit	FALSE	FALSE	TRUE
children_aboard	FALSE	FALSE	FALSE
nationality_crew	norwegian	lithuanian	norwegian
goal	1. Locate lost vessel 2. Locate lost persons	1. Manoeuvre ship 2. Get the ship loose 3. Save the vessel 4. Get water out of the boat 5. Save the environment 6. Evacuate the crew 7. Save the people	1. Put out fire 2. Evacuate the crew 3. Save the environment 4. Save the people 5. Save the vessel
action	1.1 Send nearby vessels to search 2.1 Send nearby vessels to search 2.2 Send helicopter for search	1.1 Shift from autopilot to manual control 1.2 Use emergency controls 2.1 Set motor in reverse 3.1 Send for tug boat 4.1 Use bilge pump 5.1 Stop main engines 5.2 Stop fuel supply to engines 6.1 Lower the MOB boat 6.2 Lower the inflatable raft 7.1 Send nearby vessels for rescue	1.1 Alert crew 1.2 Close air supply 1.3 Use powder extinguisher 1.4 Use flush hoses 2.1 Take on the MOB-suits 2.2 Lower the MOB-boat 2.3 Jump into the sea with MOB-suits 3.1 Stop main engines 4.1 Send nearby vessels for rescue 5.1 Send patrol vessel from the coastal guard
result	1.1 Success 2.1 Partial success 2.2 Partial success	1.1 Failed 1.2 Failed 2.1 Failed 3.1 Failed 4.1 Failed 5.1 Success 5.2 Success 6.1 Failed 6.2 Success 7.1 Success	1.1 Success 1.2 Failed 1.3 Partial success 1.4 Failed 2.1 Success 2.2 Failed 2.3 Success 3.1 Success 4.1 Success 5.1 Failed

Attribute types	Fisktrans LIAB	Viking Sky	MV Full City
category	shipwreck	technical_failure	grounding
event	technical_failure, steering, navigation, drift	technical_failure, forward_momentum	environmental_damage
distress_call	mayday	mayday	none
emergency_phase	distress	distress	uncertainty
pear	person	person	environment
daylight	FALSE	TRUE	FALSE
time_of_year	0.08	0.25	0.5
words describing the weather:	*storm*	*storm*	*small storm*
rain	12	18.4	10
wind	22	25	25
cloud_cover	TRUE	TRUE	TRUE
wave_height	8	18	5
conditions_worsen	TRUE	TRUE	TRUE
longitude	67.63	63	58
lattitute	14.51	7	9
offshore	TRUE	TRUE	FALSE
trafficated	FALSE	TRUE	TRUE
range_from_shore	short	short	short
radio_contact	TRUE	TRUE	TRUE
vessel_size	57.26	228	100
weight	969	4826	15873
ais	TRUE	TRUE	TRUE
building_year	1952	2017	1995
vessel_type	cargo_ship	cruise_ship	bulk_carrier
speed	8	0	3
pob	6	1373	23
experience	experienced	none	some
training	trained	none	some
mob_suit	TRUE	FALSE	_unknown_
children_aboard	FALSE	FALSE	FALSE
nationality_crew	norwegian	mixture	englishmen
goal	1. Keep the bow against the weather 2. Manoeuvre ship 3. Save the vessel 4. Limit drift 5. Evacuate the crew 6. Save the crew	1. Restore Power 2. Limit drift 3. Evacuate passengers 4. Save the people 5. Save the vessel	1. Limit drift 2. Get water out of the boat 3. Save the vessel 4. Save the environment 5. Save the people
action	1.1 Use front and rear side propellers 2.1 Shift from autopilot to manual control 2.2 Use emergency controls 3.1 Send rescue service vessel 3.2 Send patrol vessel from the coastal guard. 4.1 Put out anchor 5.1 Take on the MOB-suits 5.2 Jump into the sea with MOB-suits 6.1 Send helicopter for rescue	1.1 Check all alarms for problems connected to engine control room 2.1 Put out anchor 3.1 Trigger evacuation alarm 3.2 Take on lifejackets 3.3 Use lifeboats 4.1 Send helicopter for rescue 4.2 Send nearby vessels for rescue 5.1 Send for tug boat	1.1 Start main engine 1.2 Put out anchor 2.1 Use bilge pump 3.1 Send for tug boat 4.1 Contact coastal guard 4.2 Put out booms 5.1 Send helicopter for rescue
result	1.1 Failed 2.1 Failed 2.2 Failed 3.1 Failed 3.2 Failed 4.1 Failed 5.1 Success 5.2 Success 6.1 Success	1.1 Failed 2.1 Partial success 3.1 Success 3.2 Success 3.3 Failed 4.1 Success 4.2 Failed 5.1 Partial success	1.1 Failed 1.2 Failed 2.1 Failed 3.1 Failed 4.1 Success 4.2 Success 5.1 Success

Attribute types	**Chanko**	**Lill-Anne**	**Santana**
category	grounding	missing_vessel/*shipwreck*	shipwreck
event	environmental_damage, technical_failure, drift	fatality, missing_vessel, capsizing, unsecured_cargo	fatality, missing_vessel
distress_call	none	mayday	overdue
emergency_phase	uncertainty	distress	alert
pear	environment	assets	person
daylight	FALSE	FALSE	TRUE
time_of_year	0.33	0.25	0.66
words describing the weather:			
rain	0	0	0
wind	26.7	16	10
cloud_cover	_unknown_	TRUE	FALSE
wave_height	10	3	0.5
conditions_worsen	TRUE	FALSE	TRUE
longitude	69.63	67.12	58.46
lattitute	17.82	12.01	5.82
offshore	TRUE	TRUE	TRUE
trafficated	FALSE	FALSE	TRUE
range_from_shore	short	medium	medium
radio_contact	FALSE	FALSE	FALSE
vessel_size	26.24	9.4	7.85
weight	145	140	123
ais	TRUE	FALSE	FALSE
building_year	1961	1972	1992
vessel_type	tug_boat	fishing_vessel	fishing_vessel
speed	5.2	7.5	_unknown_
pob	4	1	1
experience	some	some	some
training	some	some	some
mob_suit	TRUE	FALSE	FALSE
children_aboard	FALSE	FALSE	FALSE
nationality_crew	norwegian	norwegian	norwegian
goal	1. Fix motor failure 2. Limit drift 3. Keep the bow against the weather 4. Evacuate the crew 5. Save the people	1. Locate lost vessel 2. Locate lost persons 3. Save the vessel	1. Locate lost vessel 2. Locate lost persons
action	1.1 Check engine room for problems 1.2 Stop main engine 2.1 Put out anchor 3.1 Use front and rear side propellers 4.1 Take on the MOB-suits 4.2 Lower the MOB-boat 4.2 Lower working boat 5.1 Send rescue service vessel	1.1 Send nearby vessels to search 1.2 Send helicopter for search 2.1 Send helicopter for search 3.1 Send nearby vessels for help	1.1 Send nearby vessels to search 1.2 Send local fire-fighter boat. 1.3 Send coastal guard. 1.4 Send rescue services 1.5 Send helicopter for search 2.1 Send helicopter for search
result	1.1 Failed 1.2 Success 2.1 Failed 3.1 Partial success 4.1 Success 4.2 Failed 4.3 Success 5.1 Success	1.1 Success 1.2 Success 2.1 Failed 3.1 Failed	1.1 Failed 1.2 Failed 1.3 Failed 1.4 Failed 1.5 Failed 2.1 Failed

A.2 Cases as Stored in Case Base

The cases included in this section only includes cases where the goal is to *limit_drift*, *evacuate_the_crew*, *fix_motor_failure* or *get_control_of_ship*. All of the cases are based on Appendix A.1.

incident_id	category	event	distress_call	emergency_phase	pear	daylight	time_of_year	rain	temp_air	wind	cloud_cover	wave_height
0	shipwreck	steering	panpan	distress	asset	FALSE	0.08	12	3	24	TRUE	8.0
1	shipwreck	steering	panpan	distress	asset	FALSE	0.08	12	3	24	TRUE	8.0
2	shipwreck	drift	mayday	distress	asset	FALSE	0.08	12	3	24	TRUE	8.0
3	shipwreck	grounding	mayday	distress	people	FALSE	0.08	12	3	24	TRUE	8.0
4	shipwreck	grounding	mayday	distress	people	FALSE	0.08	12	3	24	TRUE	8.0
5	technical_failure	forward_momentum	mayday	distress	asset	TRUE	0.25	18.4	-1.6	25	TRUE	18.0
6	technical_failure	drift	mayday	distress	people	TRUE	0.25	18.4	-1.6	25	TRUE	18.0
7	technical_failure	drift	mayday	distress	people	TRUE	0.25	18.4	-1.6	25	TRUE	18.0
8	grounding	drift	none	uncertanity	asset	FALSE	0.5	10	15	25	TRUE	5.0
9	grounding	drift	none	uncertanity	asset	FALSE	0.5	10	15	25	TRUE	5.0
10	grounding	forward_momentum	none	uncertanity	asset	FALSE	0.33	0	2	26.7	_unknown_	10.0
11	grounding	forward_momentum	none	uncertanity	asset	FALSE	0.33	0	2	26.7	_unknown_	10.0
12	grounding	drift	panpan	alert	asset	FALSE	0.33	0	2	26.7	_unknown_	10.0
13	grounding	grounding	mayday	distress	people	FALSE	0.33	0	2	26.7	_unknown_	10.0
14	grounding	grounding	mayday	distress	people	FALSE	0.33	0	2	26.7	_unknown_	10.0
15	grounding	grounding	mayday	distress	people	FALSE	0.33	0	2	26.7	_unknown_	10.0
16	shipwreck	steering	none	none	asset	FALSE	0.25	0	7	8.2	FALSE	0.2
17	shipwreck	steering	none	none	asset	FALSE	0.25	0	7	8.2	FALSE	0.2
18	shipwreck	grounding	none	none	people	FALSE	0.25	0	7	8.2	FALSE	0.2
19	shipwreck	grounding	none	none	people	FALSE	0.25	0	7	8.2	FALSE	0.2

incident_id	conditions_worsen	longitude	lattitute	offshore	trafficated	range_from_shore	radio_contact	vessel_size	weight	ais	building_year	vessel_type	speed
0	TRUE	67.63	14.51	TRUE	FALSE	short	TRUE	57.26	969	TRUE	1952	cargo_ship	8
1	TRUE	67.63	14.51	TRUE	FALSE	short	TRUE	57.26	969	TRUE	1952	cargo_ship	8
2	TRUE	67.63	14.51	TRUE	FALSE	short	TRUE	57.26	969	TRUE	1952	cargo_ship	6
3	TRUE	67.63	14.51	TRUE	FALSE	short	TRUE	57.26	969	TRUE	1952	cargo_ship	6
4	TRUE	67.63	14.51	TRUE	FALSE	short	TRUE	57.26	969	TRUE	1952	cargo_ship	6
5	TRUE	63	7	TRUE	TRUE	short	TRUE	228	4826	TRUE	2017	cruise_ship	5
6	TRUE	63	7	TRUE	TRUE	short	TRUE	228	4826	TRUE	2017	cruise_ship	3
7	TRUE	63	7	TRUE	TRUE	short	TRUE	228	4826	TRUE	2017	cruise_ship	3
8	TRUE	58	9	FALSE	FALSE	short	TRUE	100	15873	TRUE	1995	bulk_carrier	3
9	TRUE	58	9	FALSE	FALSE	short	TRUE	100	15873	TRUE	1995	bulk_carrier	3
10	TRUE	69.63	17.82	TRUE	FALSE	short	FALSE	26.24	145	TRUE	1961	tug_boat	5.2
11	TRUE	69.63	17.82	TRUE	FALSE	short	FALSE	26.24	145	TRUE	1961	tug_boat	5.2
12	TRUE	69.63	17.82	TRUE	FALSE	short	FALSE	26.24	145	TRUE	1961	tug_boat	4
13	TRUE	69.63	17.82	TRUE	FALSE	short	FALSE	26.24	145	TRUE	1961	tug_boat	4
14	TRUE	69.63	17.82	TRUE	FALSE	short	FALSE	26.24	145	TRUE	1961	tug_boat	4
15	TRUE	69.63	17.82	TRUE	FALSE	short	FALSE	26.24	145	TRUE	1961	tug_boat	4
16	FALSE	58.08	6.82	FALSE	FALSE	short	FALSE	30.21	285	TRUE	1984	tug_boat	8.4
17	FALSE	58.08	6.82	FALSE	FALSE	short	FALSE	30.21	285	TRUE	1984	tug_boat	8.4
18	FALSE	58.08	6.82	FALSE	FALSE	short	FALSE	30.21	285	TRUE	1984	tug_boat	8.4
19	FALSE	58.08	6.82	FALSE	FALSE	short	FALSE	30.21	285	TRUE	1984	tug_boat	8.4

incident_id	pob	experience	training	mob_suit	children_aboard	nationality_crew	goals	action	result
0	6	experienced	trained	TRUE	FALSE	norwegian	get_control_of_ship	shift_from_autopilot_to_manual_control	failed
1	6	experienced	trained	TRUE	FALSE	norwegian	get_control_of_ship	use_emergency_controls	failed
2	6	experienced	trained	TRUE	FALSE	norwegian	limit_drift	put_out_anchor	success
3	6	experienced	trained	TRUE	FALSE	norwegian	evacuate_the_crew	take_on_the_mob_suits	success
4	6	experienced	trained	TRUE	FALSE	norwegian	evacuate_the_crew	jump_in_sea_with_mob_suits	success
5	1373	none	none	FALSE	FALSE	mixture	fix_motor_failure	check_engine_room_for_problems	failed
6	1373	none	none	FALSE	FALSE	mixture	limit_drift	put_out_anchor	success
7	1373	none	none	FALSE	FALSE	mixture	limit_drift	start_main_engine	success
8	23	some	some	_unknown_	FALSE	englishmen	limit_drift	start_main_engine	failed
9	23	some	some	_unknown_	FALSE	englishmen	limit_drift	put_out_anchor	success
10	4	some	some	TRUE	FALSE	norwegian	fix_motor_failure	check_engine_room_for_problems	failed
11	4	some	some	TRUE	FALSE	norwegian	fix_motor_failure	stop_main_engine	success
12	4	some	some	TRUE	FALSE	norwegian	limit_drift	put_out_anchor	success
13	4	some	some	TRUE	FALSE	norwegian	evacuate_the_crew	take_on_the_mob_suits	success
14	4	some	some	TRUE	FALSE	norwegian	evacuate_the_crew	lower_the_mob_boat	failed
15	4	some	some	TRUE	FALSE	norwegian	evacuate_the_crew	lower_working_boat	success
16	3	some	some	FALSE	FALSE	lithuanian	get_control_of_ship	shift_from_autopilot_to_manual_control	failed
17	3	some	some	FALSE	FALSE	lithuanian	get_control_of_ship	use_emergency_controls	failed
18	3	some	some	FALSE	FALSE	lithuanian	evacuate_the_crew	lower_the_mob_boat	failed
19	3	some	some	FALSE	FALSE	lithuanian	evacuate_the_crew	lower_inflatable_raft	success

A.3 Goal Structure

- **Goal**: Manoeuvre ship
  - **Sub-goal**: Keep the bow against the weather
    - **Action**: *Use front and rear side propellers*
  - **Sub-goal**: Get the ship loose
    - **Action**: *Set motor in reverse*
  - **Sub-goal**: Get control of ship
    - **Action**: *Shift from autopilot to manual control*
    - **Action**: *Use emergency controls*
- **Goal**: Technical failure
  - **Sub-goal**: Limit drift
    - **Action**: *Put out anchor*
    - **Action**: *Start main engine.*
  - **Sub-goal**: Restore power
    - **Action**: *Check all alarms for problems connected to engine control room.*
  - **Sub-goal**: Fix motor failure
    - **Action**: *Check engine room for problems.*
    - **Action**: *Stop main engine.*
- **Goal**: Get in contact with overdue vessel
  - **Sub-goal**: Locate lost vessel
    - **Action:** *Send nearby vessels to search*
    - **Action:** *Send local fire-fighter boat.*
    - **Action:** *Send coastal guard.*
    - **Action:** *Send rescue services*
    - **Action**: *Send helicopter for search*
  - **Sub-goal**: Locate lost persons
    - **Action**: *Send nearby vessels to search*
    - **Action**: *Send helicopter for search*
- **Goal:** Save the people
  - **Action**: *Send helicopter for rescue.*
  - **Action**: *Send nearby vessels for rescue.*
  - **Action**: *Send rescue service vessel*

- **Goal:** Evacuate the crew
  - **Action:** *Take on the MOB-suits.*
  - **Action:** *Jump into the sea with MOB-suits.*
  - **Action**: *Lower the MOB-boat.*
  - **Action**: *Lower inflatable raft.*
  - **Action**: *Lower working boat.*
- **Goal:** Evacuate passengers
  - **Action**: *Trigger evacuation alarm.*
  - **Action**: *Take on lifejackets.*
  - **Action**: *Use lifeboats.*
- **Goal:** Save the vessel
  - **Sub-goal**: Put out fire
    - **Action**: *Alert crew.*
    - **Action**: *Close air supply.*
    - **Action**: *Use powder extinguisher.*
    - **Action**: *Use flush hoses.*
    - **Action:** *Send patrol vessel from the coastal guard.*
  - **Sub-goal**: Save asset
    - **Action:** *Send rescue service vessel.*
    - **Action:** *Send patrol vessel from the coastal guard.*
    - **Action**: *Send for tug boat.*
    - **Action**: *Send nearby vessels for help.*
  - **Sub-goal**: Get water out of the boat.
    - **Action:** *Use bilge pump.*
    - **Action:** *Close latches.*
- **Goal**: Save the environment
  - **Action:** *Stop main engines.*
  - **Action:** *Stop fuel supply to engines.*
  - **Action:** *Contact coastal guard.*
  - **Action:** *Put out booms*

A.4 Local Similarity Measures

The local similarity function for each attribute can be found here. The included local similarity functions are snapshots from myCBR Workbench.

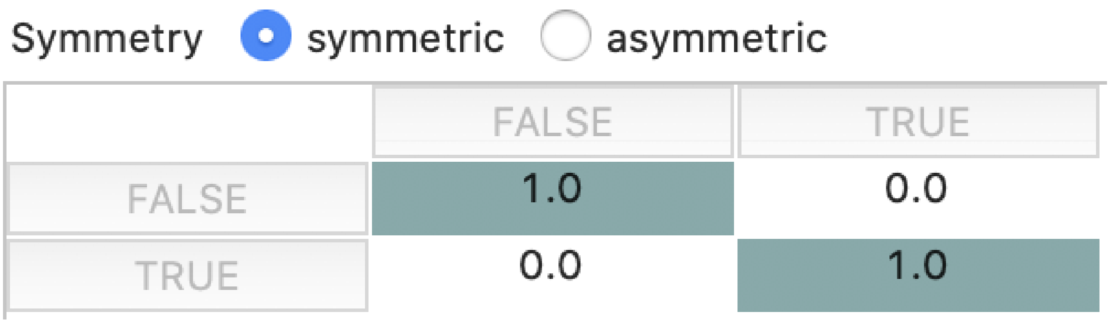

Figure A.1: Local similarity function for Boolean attributes

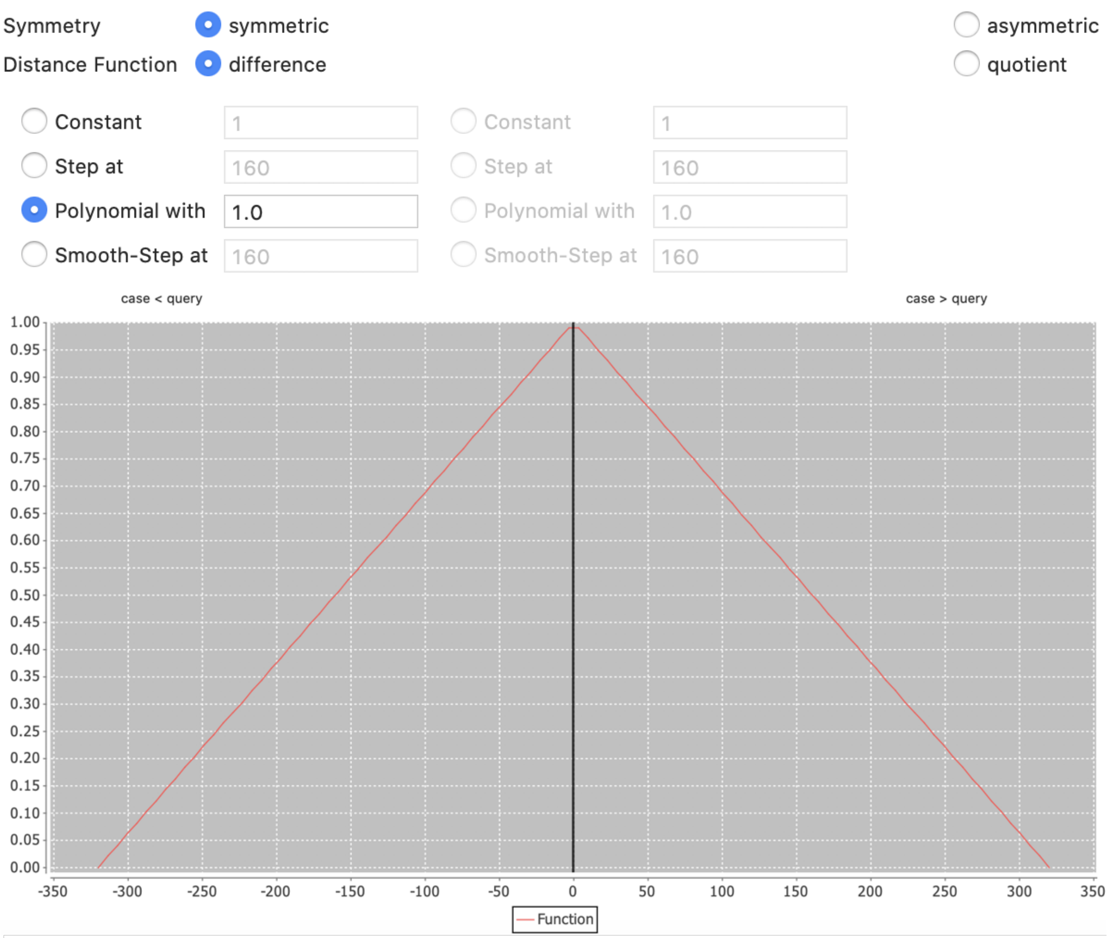

Figure A.2: Local similarity function for *building year*

Symmetry symmetric asymmetric

	experienced	none	some
experienced	1.0	0.0	0.5
none	0.0	1.0	0.3
some	0.5	0.3	1.0

Figure A.3: Local similarity function for *experience*

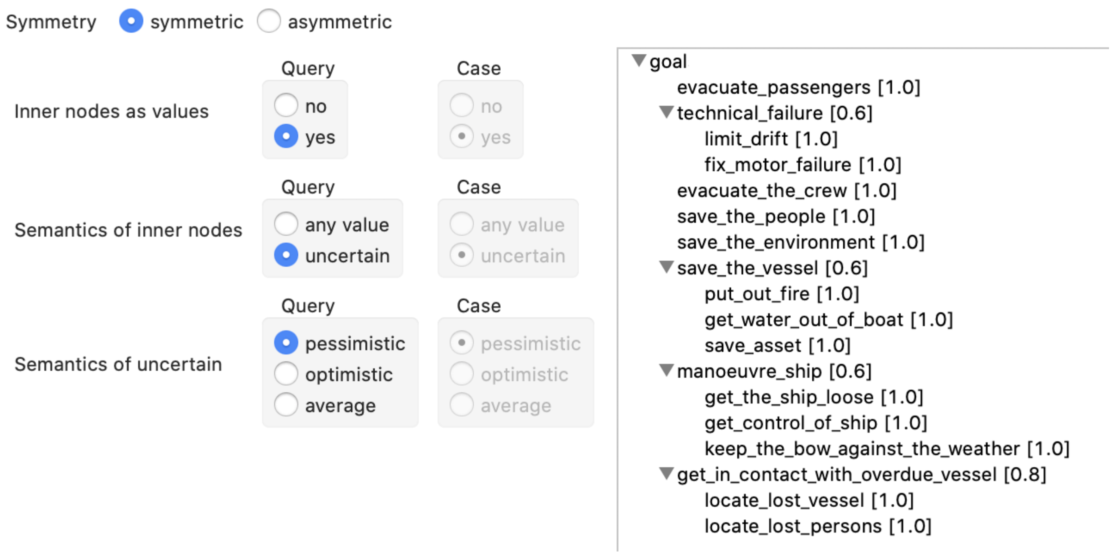

Figure A.4: Local similarity function for *goal*

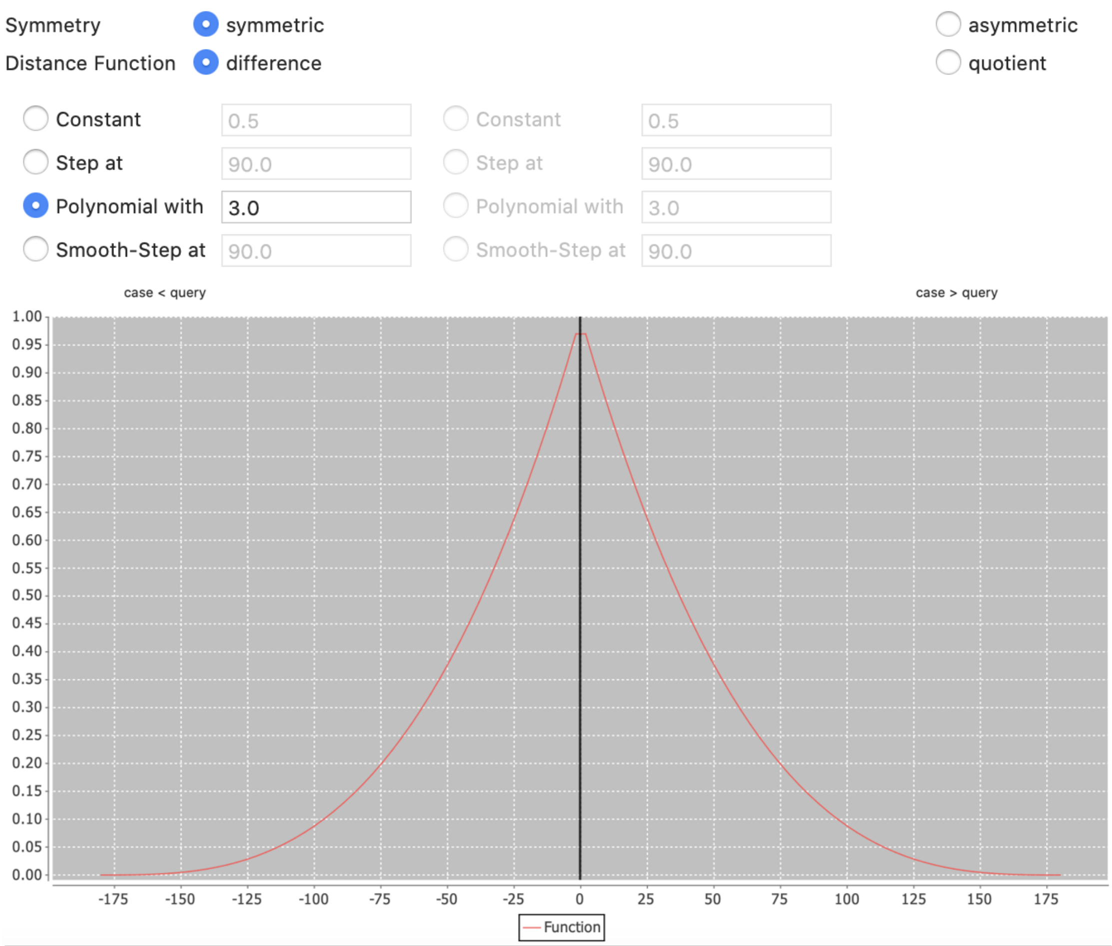

Figure A.5: Local similarity function for *latitude*

Symmetry symmetric asymmetric

	drifting_small_ves	grounding	other_accident	shipwreck	technical_failure
drifting_small_ves	1.0	0.2	0.4	0.2	0.4
grounding	0.2	1.0	0.4	0.8	0.5
other_accident	0.4	0.4	1.0	0.5	0.4
shipwreck	0.2	0.8	0.5	1.0	0.5
technical_failure	0.4	0.5	0.4	0.5	1.0

Figure A.6: Local similarity function for *category*

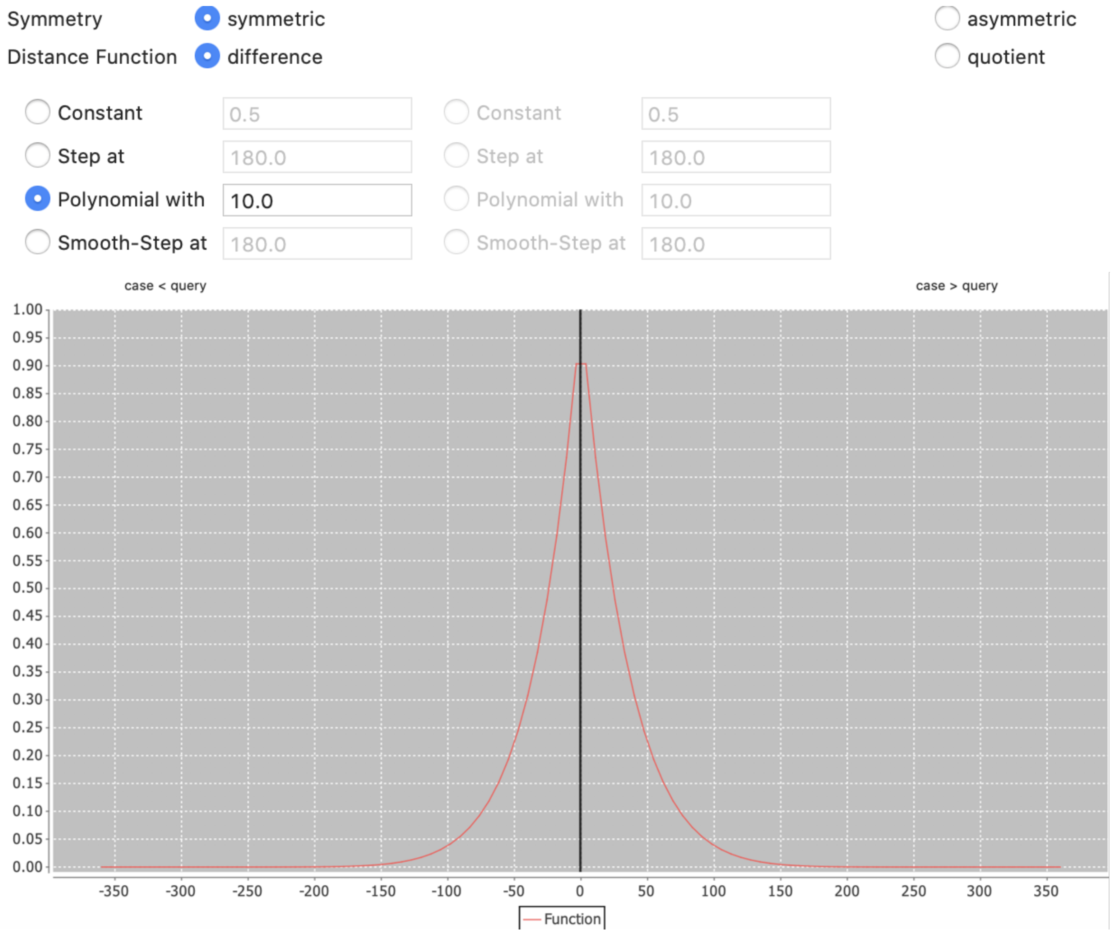

Figure A.7: Local similarity function for *longitude*

Symmetry (•) symmetric ( ) asymmetric

	drift	forward_moment	grounding	shipwreck	steering	technical_failure
drift	1.0	0.6	0.2	0.2	0.5	0.6
forward_moment	0.6	1.0	0.2	0.2	0.6	0.6
grounding	0.2	0.2	1.0	0.6	0.2	0.2
shipwreck	0.2	0.2	0.6	1.0	0.2	0.2
steering	0.5	0.6	0.2	0.2	1.0	0.6
technical_failure	0.6	0.6	0.2	0.2	0.6	1.0

Figure A.8: Local similarity function for *event*

Symmetry ◉ symmetric ○ asymmetric

	alert	distress	none	uncertanity
alert	1.0	0.6	0.0	0.6
distress	0.6	1.0	0.0	0.2
none	0.0	0.0	1.0	0.2
uncertanity	0.6	0.2	0.2	1.0

Figure A.9: Local similarity function for *emergency phase*

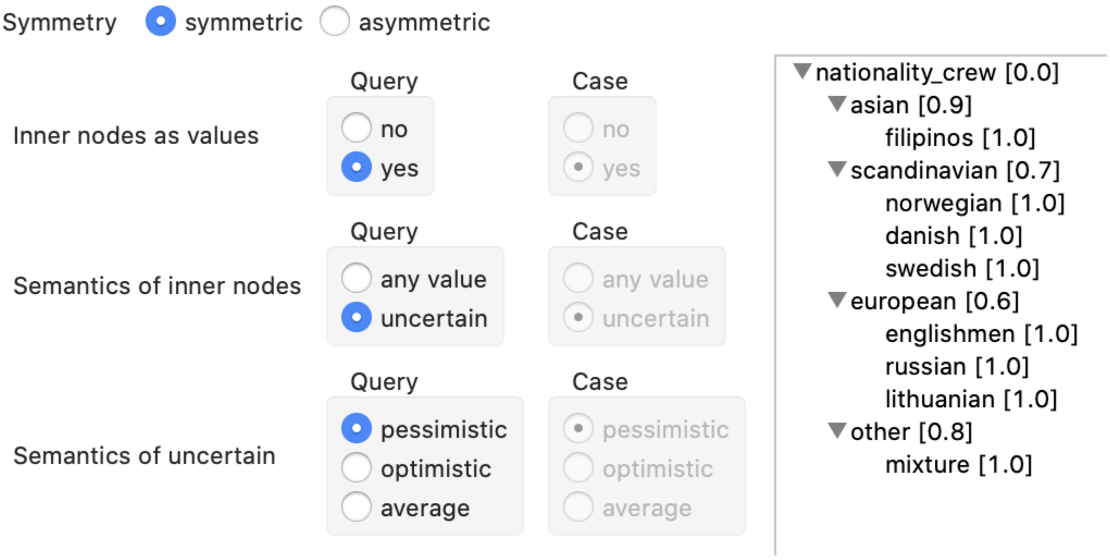

Figure A.10: Local similarity function for *nationality crew*

Symmetry ◉ symmetric ○ asymmetric

	asset	environment	people	reputation
asset	1.0	0.5	0.6	0.5
environment	0.5	1.0	0.4	0.2
people	0.6	0.4	1.0	0.1
reputation	0.5	0.2	0.1	1.0

Figure A.11: Local similarity function for *PEAR*

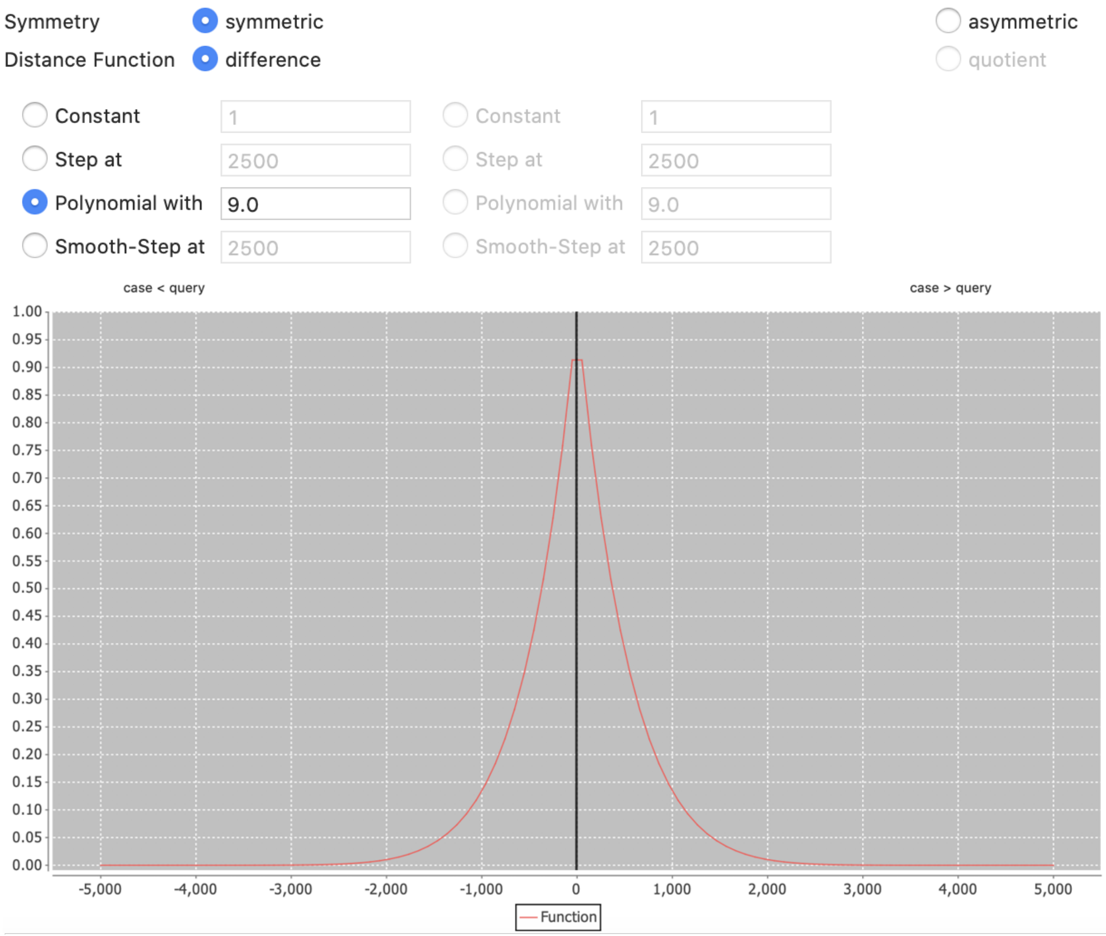

Figure A.12: Local similarity function for *POB*

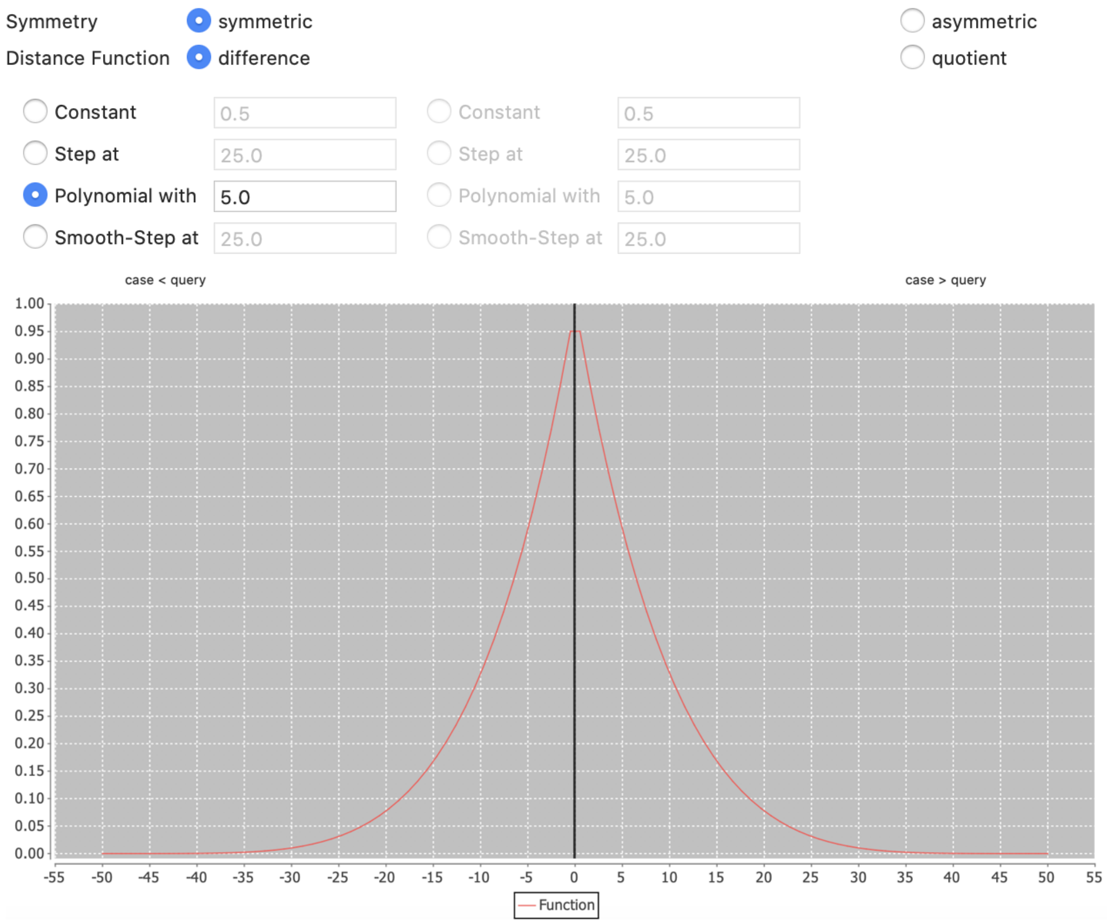

Figure A.13: Local similarity function for *rain*

Symmetry ● symmetric ○ asymmetric

	long	medium	short
long	1.0	0.7	0.1
medium	0.7	1.0	0.5
short	0.1	0.5	1.0

Figure A.14: Local similarity function for *range from shore*

v

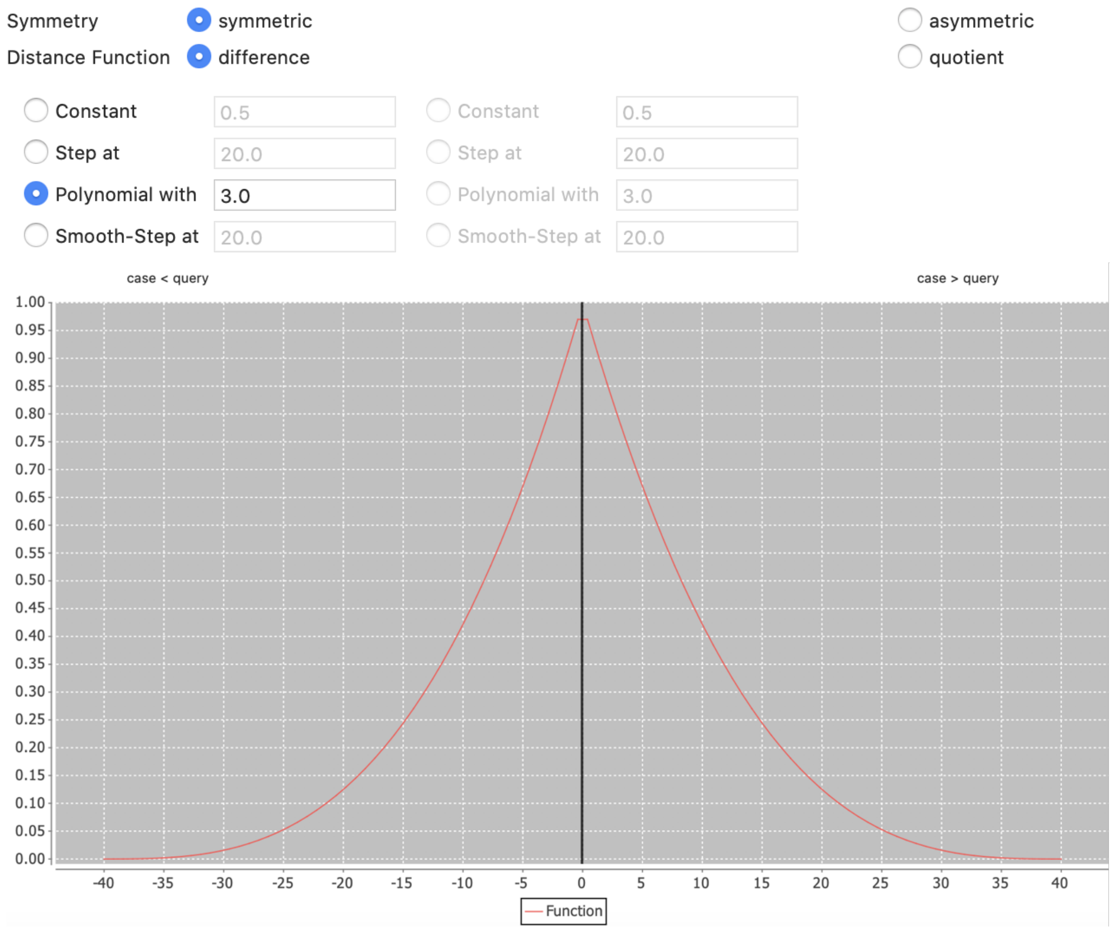

Figure A.15: Local similarity function for *speed*

Symmetry symmetric asymmetric

	mayday	none	other	overdue	panpan
mayday	1.0	0.2	0.2	0.6	0.8
none	0.2	1.0	0.2	0.2	0.2
other	0.2	0.2	1.0	0.2	0.2
overdue	0.6	0.2	0.2	1.0	0.4
panpan	0.8	0.2	0.2	0.4	1.0

Figure A.16: Local similarity function for *distress call*

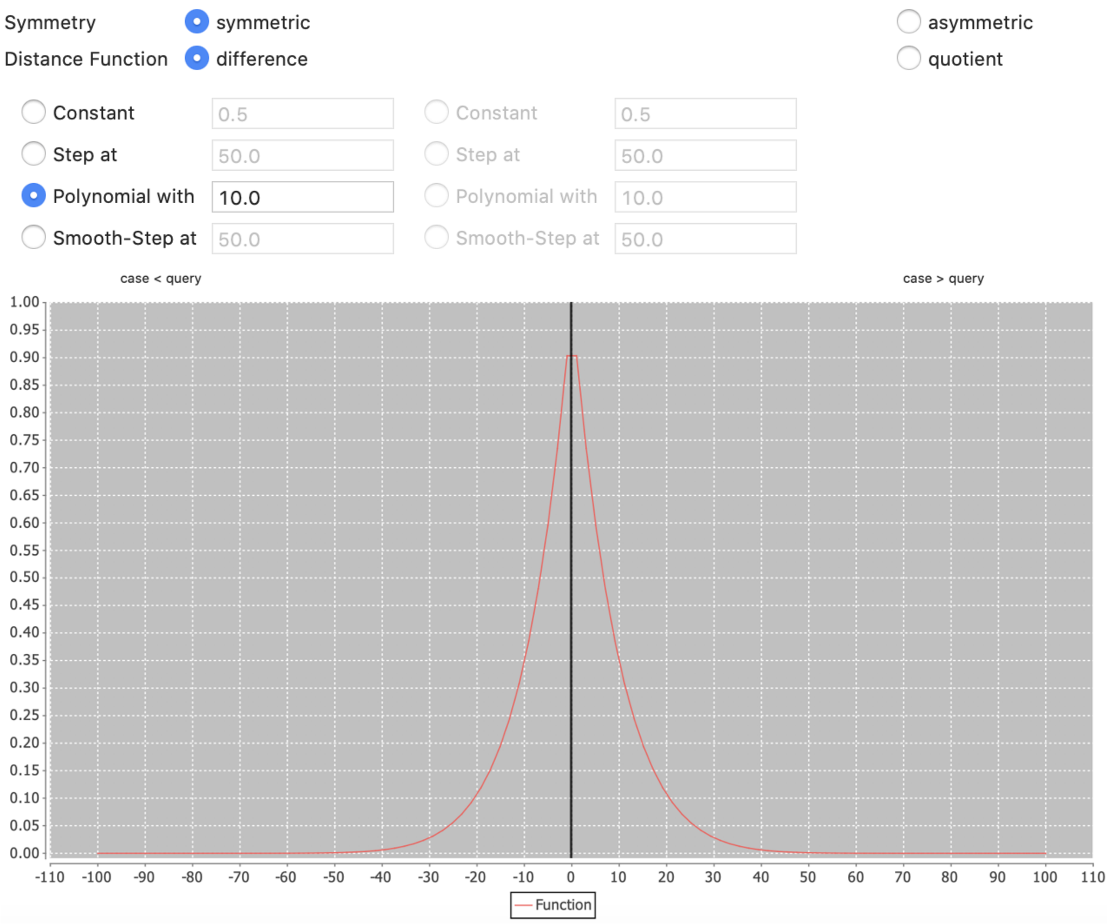

Figure A.17: Local similarity function for *temp air*

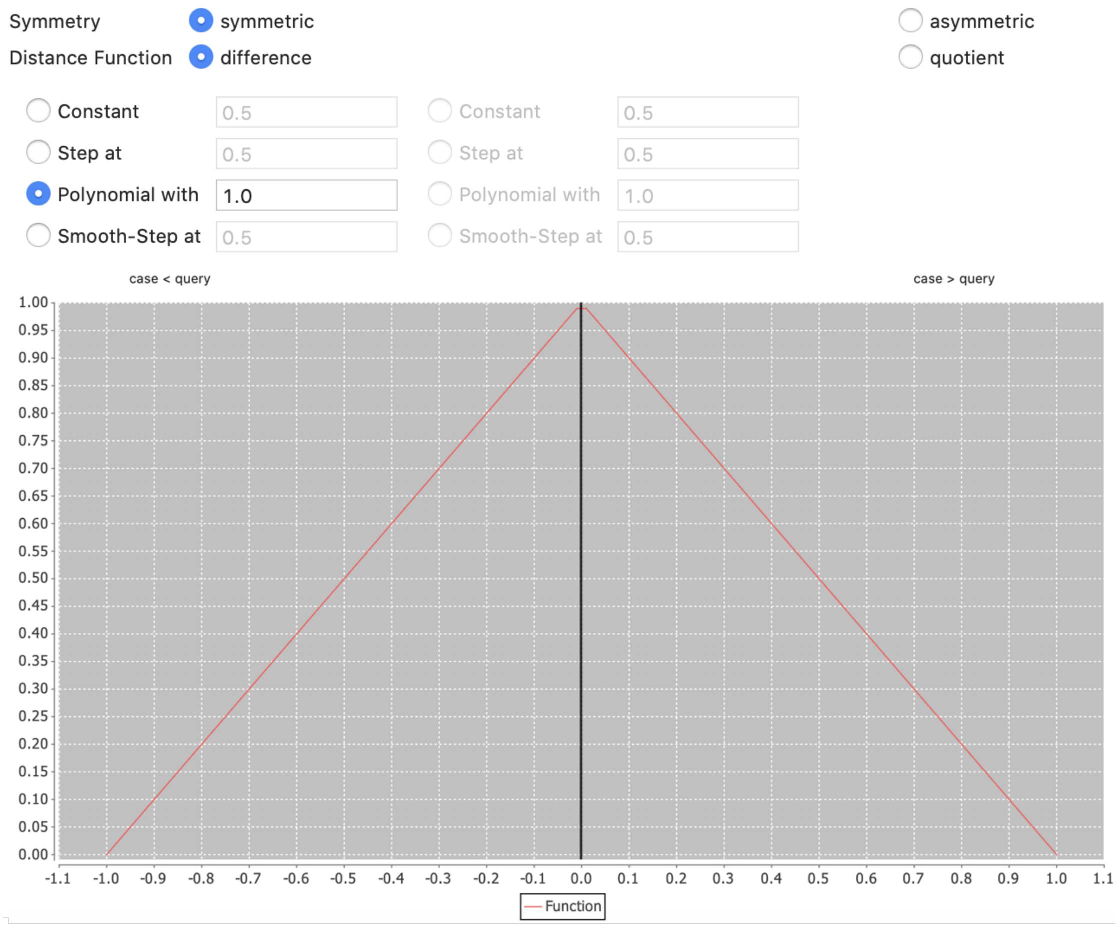

Figure A.18: Local similarity function for *time of year*

Symmetry ◉ symmetric ○ asymmetric

	none	some	trained
none	1.0	0.3	0.0
some	0.3	1.0	0.5
trained	0.0	0.5	1.0

Figure A.19: Local similarity function for training

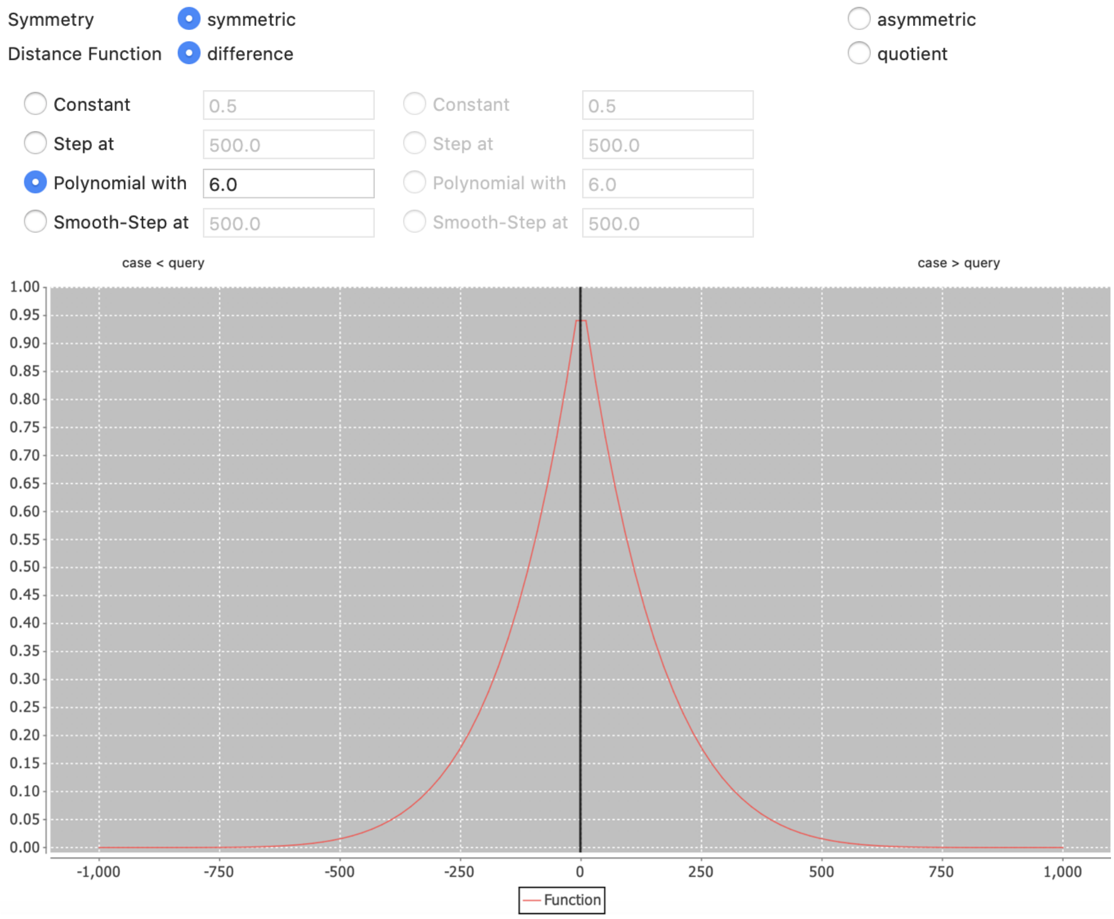

Figure A.20: Local similarity function for *vessel size*

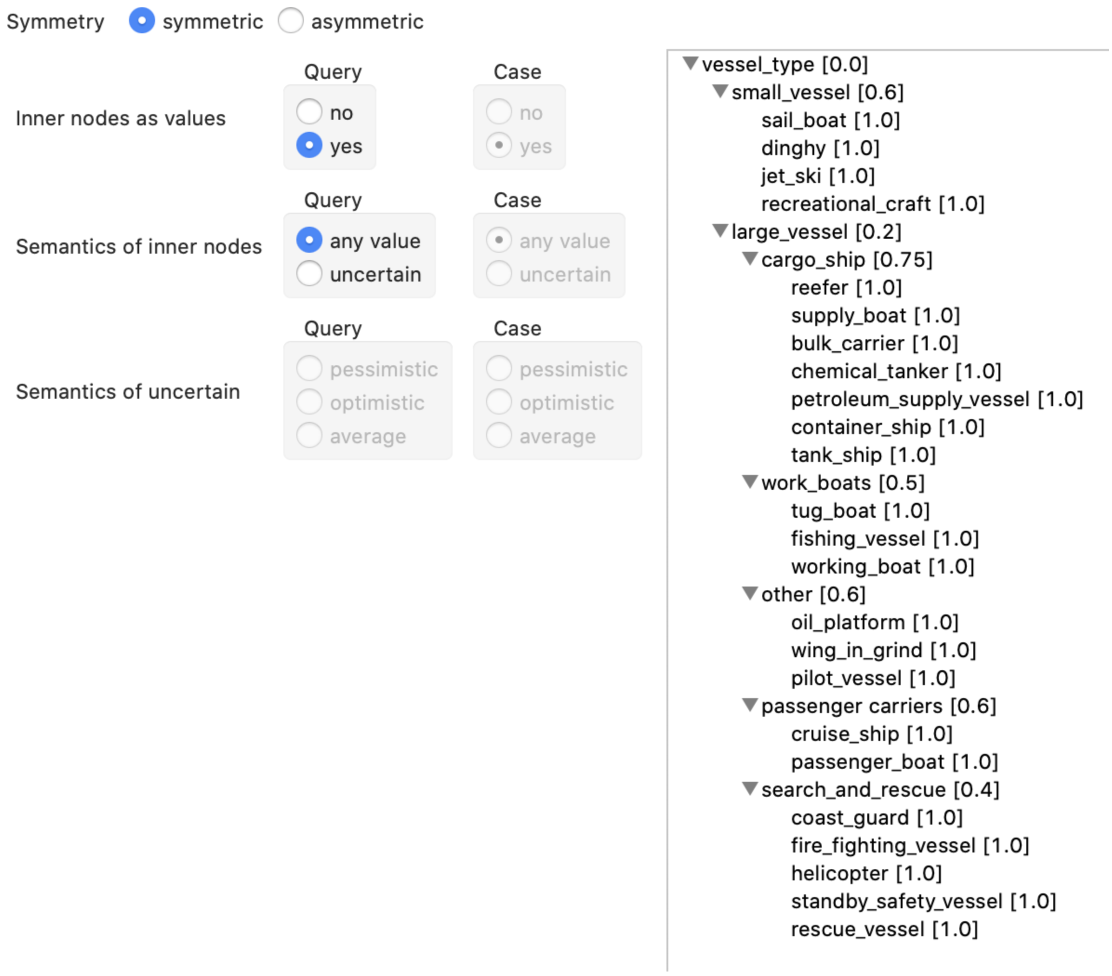

Figure A.21: Local similarity function for *vessel type*

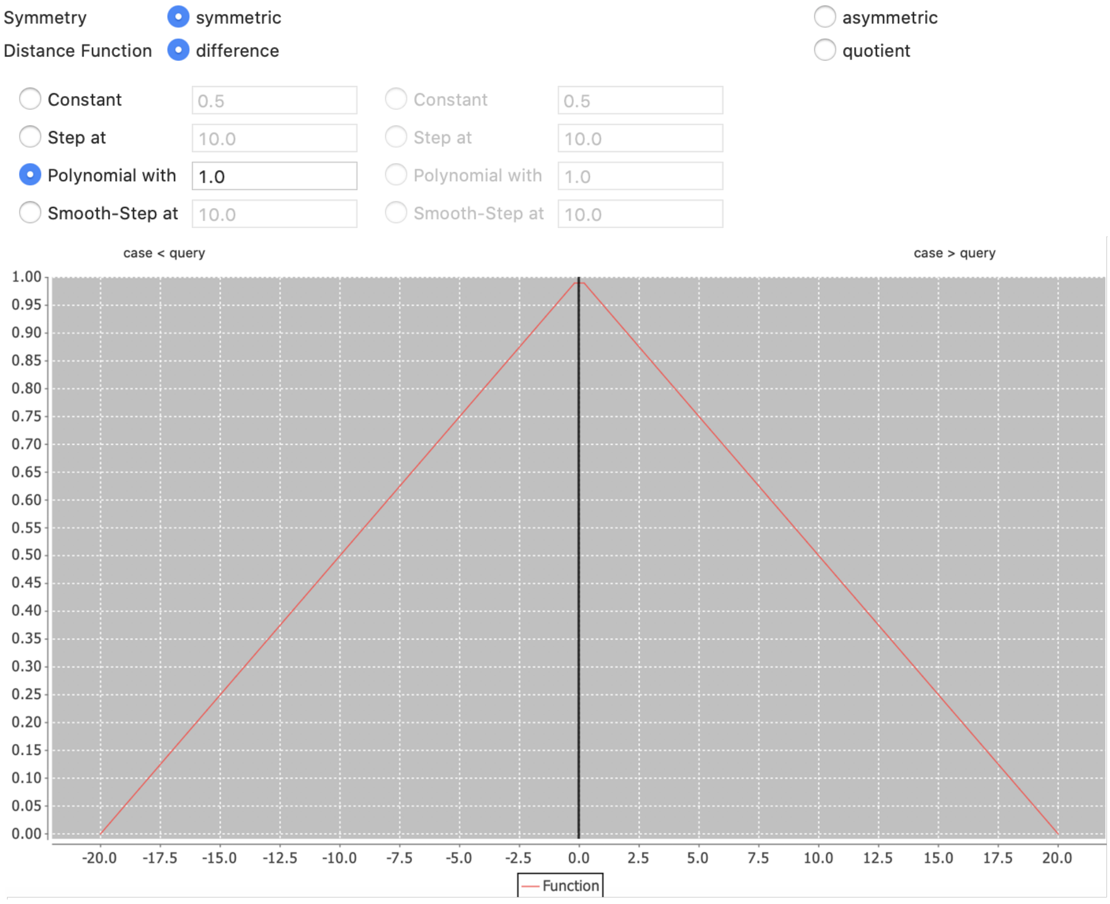

Figure A.22: Local similarity function for *wave height*

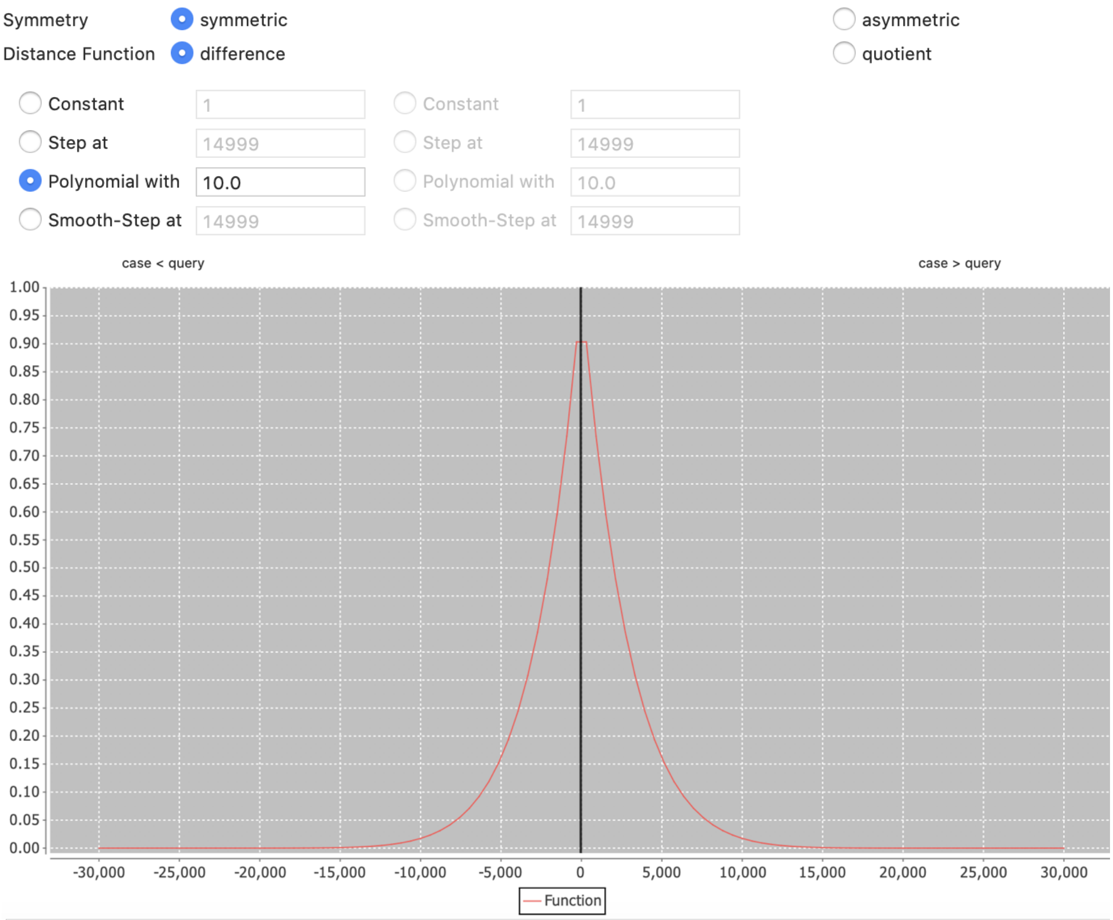

Figure A.23: Local similarity function for *weight*

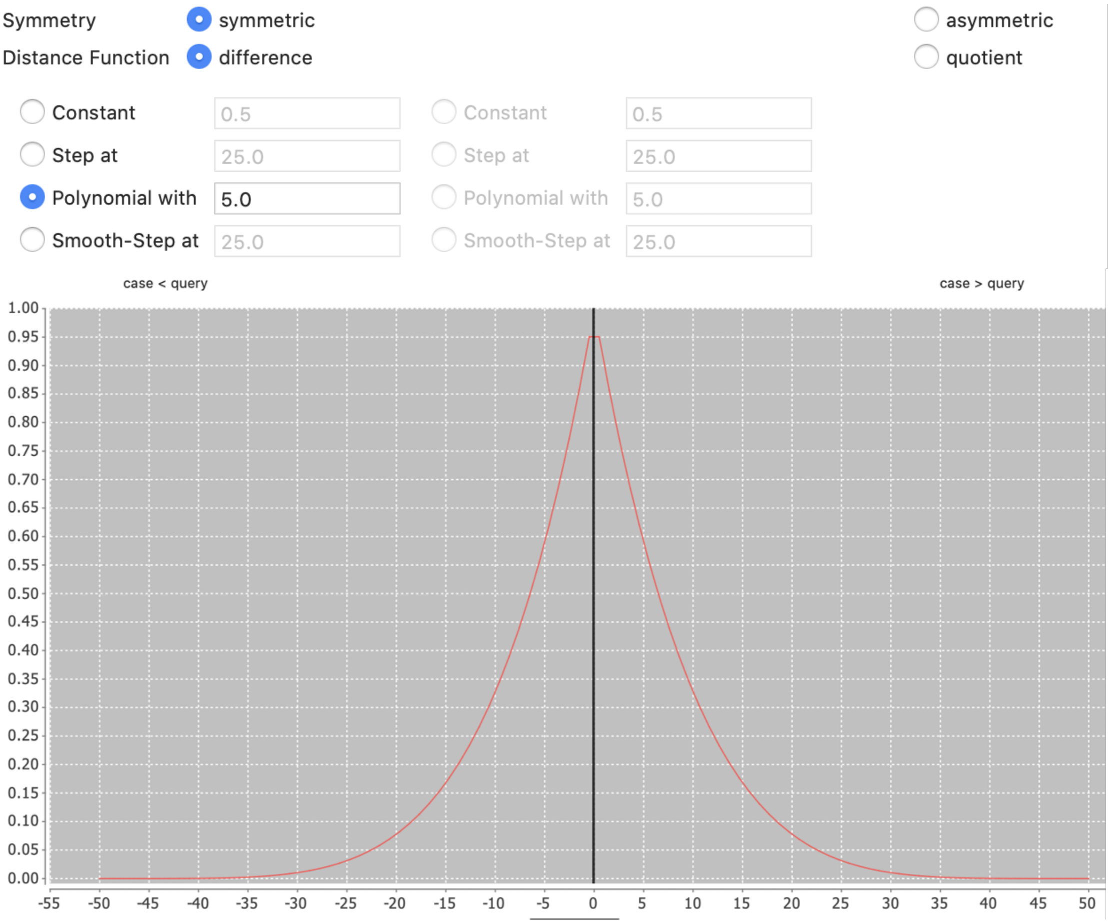

Figure A.24: Local similarity function for *wind*

Appendix B

User manual

B.1 User Manual for the Developed Prototype

User manual for the CBR-component for aiSAR

This appendix will introduce the user manual for setting up the developed prototype. In order to follow this tutorial a zip file called *aisar.zip* is needed. The *aisar.zip* file contains the following directories:

- aisar-python
- mycbr-rest-feature-enhancement-amar2
- mycbr-sdk-master
- mycbr-workbench
- orange_analysis

Python and Java are required to follow this user manual and the guide has been written based on macOS. This project has been run and tested with Java 13.0.01. First, you should install Jupyter Notebook:

https://jupyter.org/install

The zip file contains two folders holding the myCBR SDK called **mycbr-sdk-master** and the myCBR REST API called **mycbr-feature-enhancement-amar2**. These directories have been copied from the source code that can be found at *https://github.com/ntnu-ai-lab/mycbr-sdk* and *https://github.com/ntnu-ai-lab/mycbr-rest/tree/master*, where we used the branch called *feature-enhancement-amar* at the time of writing. However, since we have created a new endpoint for the myCBR REST API, edits of both the REST API and SDK were needed, and so those edits are only available in the copy included in the zip-file.

The folder **orange_analysis** contains the workflow for Orange 3 that finds the scores given by various feature relevance scoring methods. These ranks are accessible in **orange_analysis** as a file called *ranks.csv*. If you want to modify the workflow Orange 3 is needed:

https://orange.biolab.si/download/#macos

The folder **mycbr-workbench** contains aisar.prj, which is the file that contains the data modeling including the case representation, all of the cases in Appendix A.2, and local similarity measures for all attributes. The aisar.prj file can be edited using the myCBR Workbench, but as this is not a requirement for running the prototype a guide on installing myCBR Workbench has not been included.

The working product for testing the CBR-component is contained in the **aisar-python** folder. This folder contains three Jupyter Notebooks:

1. *aiSAR_dataset-k-fold-cross-validation.ipynb*
2. *aiSAR_dataset-leave_one_out-cross_validation.ipynb*
3. *test_mycbr_py_api.ipynb*

All of the notebooks use a file called *mycbr_py_api.py* that has been copied from the myCBR REST API at *https://github.com/ntnu-ai-lab/mycbr-rest/tree/master.* This file is a wrapper for the myCBR REST API and also contains some small modifications to enable access to the new endpoint we created. The files *cross_validation.py and ranks.py,* contains all of the code that we have developed for the evaluation methods and the algorithms for finding global similarity weights. Both of these files are accessed by the Jupyter Notebooks. The third notebook has been modified from *https://github.com/ntnu-ai-lab/mycbr-rest/tree/master*.

In order to run the prototype enter the folder **mycbr-sdk-master** and run the command

mvn clean install

This command only needs to be used once, when setting up the system. Then, enter the folder **mycbr-feature-enhancement-amar2** using the terminal and run the command for setting up the system

mvn clean install

Now, everything should be set up and the following command should be run while in the folder **mycbr-feature-enhancement-amar2:**

java -DMYCBR.PROJECT.FILE=./src/main/resources/aisar.prj -jar ./target/mycbr-rest-2.0.jar

This command will start the myCBR REST API and is necessary in order to access the CBR-component. The newest version of the *aisar.prj* should always be copied into *mycbr-feature-enhancement-amar2/src/main/resources/* if it is modified by the myCBR workbench*.*

The first and second notebook contains the code for finding the best global similarity measures given the MAP and illustrates the confusion matrices, box plots and MAP-graphs. The notebooks use the REST API to access the CBR-component. In order to run the Jupyter Notebooks one needs to enter the **aisar-python** folder in the terminal and type the following command:

jupyter notebook

You need to run *aiSAR_dataset-leave_one_out-cross_validation.ipynb* before being able to run *test_mycbr_py_api.ipynb*, as one function is dependent on a global similarity measure created by the second notebook. All of the files that are necessary for the Jupyter Notebooks are available in the *aisar.zip* folder and you should now be able to test the system that we have developed.

May Helen Robertsen Storvik

Case-Based Reasoning for Decision Support in Search and Rescue

NTNU

Norwegian University of
Science and Technology