ABSTRACT

_incident reporting is becoming increasingly important in large organizations. Legislation is progressively being introduced to deal with this information. One example is the European Directive No. 94/95/EC, which obliges airlines and national bodies to collect and collate reports of incidents. Typically these organizations use manual files and standard databases to store and retrieve incident reports. However, research has established that database technology needs to be enhanced in order to deal with incidents. This paper describes the design and implementation of InRet, an incident report retrieval system that endeavours to find similarities and patterns between incidents by combining the strengths of Case-Based Reasoning and Information Retrieval techniques in an integrated system. Preliminary results from InRet are presented and are encouraging.

INTRODUCTION

The use of incident reporting systems is increasingly being recognized as an effective way of analyzing incidents, leading to the anticipation and preclusion of further incidents and accidents. An incident can be defined as "an unwanted disruption that has unfortunate and untoward consequences" (Perrow, 1999). Groups of incidents may be related in subtle and unremarkable ways. Such relationships may not be detectable using standard database queries, hence the need for advanced incident management systems (Johnson, 2000). The goal of incident reporting and analysis is to determine the links between incidents in order to prevent recurrences. This requires a technology that can extract useful information from large amounts of accumulated data.

The purpose of data mining is to look for hidden patterns in a group of data that can be used to predict future behaviour. It involves sorting through data in order to identify associations and to establish relationships. InRet uses data mining in order to detect such relationships between incidents. It does so using Case-Based Reasoning and Information Retrieval techniques.

1.1 Introduction to Case-Based Reasoning

Case-based reasoning (CBR) is an Artificial Intelligence problem-solving technique that solves new problems by reusing existing problem solutions stored in the form of cases in a case-base (Leake, 1996). The CBR process can be described in four main stages:

1. **Retrieve**: A new problem is compared/matched with the cases in the case-base to retrieve the most similar cases;
2. **Reuse**: The solution(s) to the retrieved case(s) is reused to provide a solution to the new problem;
3. **Revise**: Unless the retrieved solution(s) is an identical match, it may have to be revised;
4. **Retain**: The new case, incorporating both problem and solution, is retained in the case-base.

Each case is represented by a set of features, or attributes, which are used in the matching process. Depending on the user's preference, and on the query, these features may or may not be weighted. Weighting means that it is possible to give more importance to certain features over others.

1.2 Introduction to Cosine Similarity

The cosine similarity metric is an Information Retrieval (IR) technique which determines the similarity between documents by measuring the cosine of the angle between vectors of terms. Documents and queries can be represented by vectors of terms, such that a system selects documents in response to a query, by identifying documents whose vector representations are most similar to that of the query vector.
2 CORPUS

InRet uses a small corpus of sixty-nine aviation incident reports provided by the Air Accident Investigation Unit (AAIU) of Ireland, whose purpose is to investigate civil aviation accidents and incidents with the objective of preserving life and the avoidance of similar recurrences. This is a real-world domain, and real data is used in the experiments. Every incident report consists of a number of defined features and a free text description. The data is divided into two separate storage depositories: a case-base and a keyword index.

Each report consists of a fixed number of defined features, e.g. aircraft manufacturer, aircraft registration, date of incident, time of incident, which describe the incident. For the purpose of these preliminary experiments, twelve separate features out of a possible eighteen available features, have been extracted and stored in the case-base. This is illustrated in Figure 1.

![Figure 1: Feature Extraction from Aviation Corpus]

A set of keywords has also been determined for each incident, and these are stored in a keyword index. This allows the user to describe an incident in more detail than the defined features allow. These keywords have been taken from the free text description included in each incident report and do not include the pre-defined features in the case-base.

3 INRET SYSTEM ARCHITECTURE

InRet comprises three layers:

1. User Interface: This is built upon the operations carried out in the management layer and provides the user with a means of interacting with the system.
2. Management Layer: This layer is responsible for the storage and retrieval operations. It acts as an intermediary between the user interface and the storage layer, using the low-level data in the storage layer to answer the user's queries. InRet uses programs written in C to carry out these operations.
3. Storage Layer: This is where the data is stored. InRet's storage layer is made up of a case-base and a textual keyword index. The case-base consists of a set of cases where each case corresponds to an incident. The keyword index contains words which are considered to be important in the description of the incident but which are not included in the defined features. These keywords are taken from the free text description of the incident.

![Figure 2: InRet System Architecture]

The three layers while independent of each other, interact and communicate in order to provide a functional incident management system. Each layer's independence allows alterations to be made within the layers without affecting the overall functionality of the system.

4 OVERVIEW OF INRET

In any organization dealing with incident reports, it is standard practice to compare all new incoming incident reports with those already stored, to check if similar incidents have previously occurred. Such a procedure allows an organization's safety officer to establish whether a new incident is a "once-off" or part of a pattern of incidents. A repetition of similar incidents may be due to some design or procedural problems that an organization needs to address.

The investigation at hand focuses on the problem of comparing incoming incident reports with those already stored. Because of the small corpus size, we adopted the following CBR approach:
1. Extract the features and store all incident reports in a case-base of these incidents; Take the n\textsuperscript{th} incident (initially n=1) and treat it as a new incident, i.e. compare it to all of the remaining incidents;
2. List the related incidents;
3. Repeat steps 2 and 3 for all incidents in the case-base;
4. Evaluate the performance of InRet.

The cosine similarity approach is similar. However, the keywords are stored in a textual keyword index as opposed to a case-base of features.

5 METHODOLOGY

The system should match an incident presented by the user against each incident in the corpus, and retrieve the similar incidents. Three separate techniques are used for the retrieval of similar incidents. These are CBR, cosine similarity and a combination of these two techniques.

5.1 CBR Methodology

Two separate CBR methods are used for the retrieval of similar incident reports. These are unweighted and weighted forms of matching.

According to unweighted matching, every feature is awarded an equal weight. Every feature in the incident is compared with the corresponding feature in each of the other incidents. If the features match, a score of 1 is awarded. If the features do not match, a score of 0 is awarded. A similarity score is calculated by:

1. Finding the sum of the matching features;
2. Dividing this sum by the number of features contained in the incident, as in the formula below:

\[
Sim(T, S) = \frac{\sum_{i=1}^{n} f(T_i, S_i)}{n}
\]

If the similarity score between two incidents exceeds a predetermined threshold, then they are considered to be similar. Several experiments were run in which the similarity threshold was varied in order to determine a suitable threshold value.

It may be agreed that some features are more important than others. For example, they type of aircraft may be considered to be of greater significance than the date on which the incident occurred. Subsequently, the user may decide to give more credence to the fact that the aircraft types match than to the fact that the two incidents occurred on the same day. This is provided for in the weighted matching technique, where a user can assign a score to each feature ranking the importance of each feature intuitively.

A second weighted matching technique is also used in which sets of weights are randomly picked and assigned to the features. For illustration purposes, only three of these sets, identified as Random Weights 1, Random Weights 2 and Random Weights 3, are shown. These results are compared against the results of the intuitively weighted technique.

As in the previous method, every incident is matched against each of the other incidents in the case-base. To compute the similarity between two incidents:

1. Sum the relevant scores together;
2. Divide this result by the sum of all the weights, as in the formula below:

\[
Sim(T, S) = \frac{\sum_{i=1}^{n} f(T_i, S_i) \cdot w_i}{\sum_{i=1}^{n} w_i}
\]

5.2 Cosine Similarity

This technique, derived from vector theory, calculates the extent of the similarity between the incidents by measuring the cosine of the angle between vectors of terms, or keywords. The keyword index is processed in order to produce a file which allows the retrieval system to locate, and count, for any given incident, all keywords occurring in that incident. Every incident in the keyword index is matched against every other incident in the index, using the cosine similarity metric. As in the CBR methodology, several different similarity thresholds are used to determine the most similar incidents. The formula for calculating the cosine similarity is:

\[
Cos(T, S) = \frac{\sum_{i=1}^{n} T_i S_i}{\sqrt{\sum_{i=1}^{n} T_i^2} \sqrt{\sum_{i=1}^{n} S_i^2}}
\]

5.3 Combined Methodology

The final similarity metric used in the evaluation of InRet involved the amalgamation of the previous two methods. The results of the CBR methodology and those of the cosine similarity were awarded separate weights. For each incident, the similarity was computed by:

1. Getting the product of each result (CBR result and cosine similarity result) and it's weight;
2. Summing the products together and dividing this number by 2, as follows:

\[ Sim(T, S) = \frac{(res \cdot CBR \cdot w_1) + (res \cdot Cos \cdot w_2)}{2} \]

A number of experiments were run in which the weights were varied to see if a hybrid system performed better than a system using a single methodology.

6 RESULTS/EVALUATIONS

A manual evaluation of the incidents was carried out by a single manual evaluator. This was a controlled evaluation against which we can assess the performance of the matching techniques. The "similar" incidents obtained from InRet are compared against the evaluator's incidents to produce a set of results. These results are shown in Figures 3 to 6.

Performance is measured using the following metrics: Precision, Recall, False Alarm Rate and Miss Rate.

1. **Precision** is the number of relevant incidents retrieved compared with the total number of incidents retrieved.
2. **Recall** is the number of relevant incidents retrieved compared with the total number of relevant incidents.
3. **False Alarm Rate** is the number of irrelevant incidents retrieved compared with the total number of incidents retrieved.
4. **Miss Rate** is the number of relevant incidents not retrieved compared with the total number of relevant incidents.

Both a high precision and a high recall result are desired. However, in general, as precision increases, recall decreases, and vice versa. This is a well-known trade-off in information retrieval. Similarly, as the false alarm rate decreases, the miss rate increases. Ideally, these metrics would both be low.

Rather than focusing on each metric separately, a precision versus recall curve, and a false alarm rate versus miss rate curve, was generated for each of the methodologies.

Figure 3 illustrates the performance of each of the CBR systems, and that of the cosine similarity, in terms of precision versus recall. A system with high precision and high recall performs better than one with low precision and/or low recall. Therefore, the best performing system will be the one furthest away from the origin. The superiority of the intuitively weighted system is obvious. The cosine similarity provides the worst result by far.

Figure 4 illustrates the performance of the systems with regard to the false alarm rate and the miss rate. The intuitively weighted system performs best at the lower thresholds. However, the results of Random Weights 2 and the unweighted system follow closely behind. The cosine similarity performs least well.

The set of results for one of the combined experiments is shown below. Figure 5 shows the results of the precision versus recall graphs in which the CBR and the cosine similarity results are weighted equally. At a higher threshold, the intuitively weighted system performs best, whereas the performance of Random Weights 3 deteriorates as the threshold increases. Overall, Random Weights 1 has the
worst performance. Otherwise, the systems perform similarly.

It is also observed that the intuitively weighted system generally provides a better set of results than either the randomly weighted systems or the unweighted system. Therefore we can surmise that a superior set of weights does exist. The problem lies with how to weight the features.

Comparing the CBR results with those of the combined similarity indicates that using CBR alone produces a better system than one using a combination of CBR and cosine similarity. The results for the combined similarity have been impaired by the poor results produced by the cosine similarity.

7 CONCLUSIONS

This paper describes the design and implementation of InRet, a basic incident report retrieval system. The results of the preliminary experiments, outlined above, are encouraging. The CBR results suggest that a favourable set of weights will produce better results than either randomly assigning weights or leaving the features unweighted.

The evaluations also indicate a poor performance using the cosine similarity measure. However, this is probably due to the total elimination of keywords which were already defined as features. A more comprehensive set of experiments is currently underway.

8 LIMITATIONS OF CURRENT APPROACH

One obvious limitation of our research is the small corpus size and the fact that a single human evaluator carried out the evaluation. Unfortunately, no TREC-like or TDT-like corpus exists for incident retrieval research. We are currently approaching organizations such as airlines and health authorities with a view to gaining access to a larger corpus and expert evaluators.

Weights can be assigned to the features either manually or automatically. Both methods have limitations. Without a specific query or user, it is not obvious as to how the weights should be delegated manually. An automatic weighting scheme would allow InRet to run through various sets of weights before deciding how it should allocate weights to achieve the best possible results. However, a user may wish to use different weights in relation to a specific incident.

For ease of implementation, a limited number of defined features, and keywords, were used to describe each of the incidents. Other features have been excluded for the purpose of these preliminary experiments, but will be included in future work.

9 FUTURE WORK

The system is still undergoing development and we are investigating a number of possible enhancements such as:
1. Methods for automatically computing feature weights as well as allowing users to guide the weighting process;
2. Using all features available in the incident report;
3. Allowing the textual component to include terms already included in the defined features;
4. Utilizing lexical resources such as WordNet (Fellbaum, 1998) to handle matching synonyms, e.g. so that "plane" matches "airplane".
5. Expanding the experimental corpus and improving the evaluation methodology so as to be able to generate more reliable results.

ACKNOWLEDGEMENTS

The support of the Basic Research Initiative of Enterprise Ireland is gratefully acknowledged.

REFERENCES


