**Abstract:**

Case base reasoning (CBR) is a process of solving problems based on the solution of previous, similar problems. A CBR system make use of a case library which store previous cases and use them in the evaluation and classification of new problems. Hence, these kinds of systems are easy to maintain and update. By combining a CBR system with a sound analyzer it is possible to create a system that can learn to recognize different types of sounds and classify them accordingly. By extracting the distinct features of each sound and measuring its similarity to the new problem case it is possible to produce a raw value of how likely it is that the sound belongs to a given group. Using this method we have been able to create a software program capable of recognizing a human sneeze with a high level of reliability, able to correctly classify the given samples with over 90% hit rate.

Case-Based Reasoning: Foundational Issues,

Methodological Variations, and System

Approaches (AICom)

Discussed by Simon

In this report, Agnar Aamodt and Enric Plaza explains and discuss the basic workings and history of the case based reasoning system(CBR). They explain how Case-based reasoning is a fairly new approach to problem solving and learning that has gained a lot of attention over the last few years. Originating in the US, the basic idea and underlying theories have spread to other continents, and we are now within a period of highly active research in case-based reasoning in Europe, as well. Over the last few years, case-based reasoning (CBR) has grown from a rather specific and isolated research area to a field of widespread interest. Activities are rapidly growing - as seen by the increased rate of research papers, availability of commercial products, and also reports on applications in regular use. The paper future goes on and explains in details how a CBR systems operates and what components it has. The paper provided important details and insights for this project, explaining how the different components of a CRB operates and interact with each other, assisting in solving many of the issues that came up during the design and development process.

Case-Based Reasoning and User-Generated AI for Real-Time Strategy Games(er-1-02)

Discussed by Simon

In the report, Santiago Onta˜n´on and Ashwin Ram discuss modern approaches to the use of CBR in computer games. Over the last thirty years computer games have become much more complex, offering incredibly realistic simulations of the real world. As the realism of the virtual worlds that these games emulate improves, players also expect the characters inhabiting these worlds to behave in a more realistic way. Thus, game developers are increasingly focusing on developing the intelligence of these characters. However, creating (AI) for modern computer games is both a theoretical and engineering challenge. For this reason, it is hard for end-users to customize the AI of games in the same way they currently customize graphics, sound, maps or avatars. The paper goes on to discuss how game developers may make use of a CBR system in their game in order to solve these issues and improve the quality of their game. While this paper was not as useful to the project in terms of content as some of the others, it did provide great insights in different areas of application for a CBR system. The paper also discuss the inherent problems with using a CBR inside a very broad domain, making it harder for the system to find an optimal solution to the presented problem.

6.5 Case Library maintenance

Maintaining a high quality and relevant case library is critical for the performance and efficiency of any CBR system. Having too many cases will result in slow evaluation and system slowdown. Having too few cases can lead to insufficient data resulting in a high error ratio. In order to keep the case library in optimal condition cases that are no longer relevant or inconsistent should be removed while new cases with better consistence and contribution should be added. Hence, cases that are rarely used in evaluation or causes incorrect evaluations are prime candidates for removal.

6.6 System Performance evaluation

When evaluating the case library the system strive to maintain the current ratio of sneeze/non-sneeze cases as well as the total volume of the library. As such, if we add a non-sneeze case we also make sure that we remove a non-sneeze case and vice versa. This is the process used for evaluation process:

1. Add a new, random case of known status to the library
2. Calculate the SF value for every case
3. Find the worst case by:
   * 1. Select the case which has participated in voting but voted wrong every time
     2. If no such case exist: Select the case which has never participated in voting and has the lowest SF value
     3. If no case has been selected at this point, select the case with lowest SF value
4. Remove the worst case from the case library
5. Repeat 1-4 for every candidate case available to the CBR system.