An Experimental Study of Increasing Diversity for Case-Based Diagnosis

Lu Zhang, Frans Coenen, and Paul Leng

The Department of Computer Science, The University of Liverpool, Liverpool L69 3BX, UK
{lzhang, frans, phl}@csc.liv.ac.uk

Abstract. Increasing diversity for case-based reasoning (CBR) is an issue that has recently drawn the attention of researchers in the CBR field. Several diversification techniques have been proposed and discussed in the literature. However, whether and to what extent those techniques can bring about benefits to end-users remains in question. In this paper, we report an experiment in applying a diversification technique to a case-based diagnosis tool in a product maintenance domain. The results of this offer some evidence in support of diversification techniques.

Keywords. Diagnosis, Diversification, Product maintenance

1 Introduction

Recently, the issue of increasing diversity for case-based reasoning (CBR) systems has become a focus of discussion (see e.g. [5], [1], [6], and [4] etc.). Most research on this issue has been carried out for recommendation systems. The main reason for addressing this problem is that the retrieved cases are usually similar to each other and thus can only offer very restricted choices to users [6]. However, there is insufficient evidence that end-users of the systems can substantially benefit from increasing diversity techniques.

In this paper, we report an experiment in applying an adaptation of the '*diversification by elimination*' technique proposed in [4] to a case-based diagnosis tool in a product maintenance domain. In case-based diagnosis systems, each retrieved case is used to suggest a possible fault. As different cases may indicate the same fault, eliminating some cases reporting the same fault may substantially increase the number of suggested possible faults in the retrieved cases. Therefore, for a given size of the retrieval set, the diversified retrieval set may lead to an increased probability that the real fault of the case under diagnosis is suggested among the faults of the retrieved cases. This increase may be very valuable for domains in which CBR can only achieve low accuracy.

To evaluate to what extent the diversification strategy in our tool can be beneficial, we performed some experiments on some real data acquired from a commercial domain. From our experiments, we found that, within a certain range of retrieval set sizes, the diversified approach achieved significantly greater success in including the real fault in the retrieve set than did the un-diversified approach, although neither achieved a very high success. These results suggest that diversification can be an effective technology for domains in which normal CBR approaches can only achieve low accuracy.

The remainder of this paper is organised as follows. Section 2 provides a general description of the case-based diagnosis tool. Section 3 reports the diversification technique used in the tool. Section 4 reports some experiments based on the tool using real data. Section 5 discusses some limitations of our study. Section 6 concludes this paper.

2 The Case-Based Diagnosis Tool

In this section, we describe the case-based diagnosis tool in a product maintenance domain, on which our case study for diversification has been performed.

2.1 Domain

The diagnosis problem we are facing originates from the needs of a manufacturer of domestic appliances in a flexible manufacturing context, whose name is Stoves PLC. The company concerned can deliver more than 3000 versions of its cookers to customers, making it possible to satisfy a very wide range of different customer requirements. However, this creates a problem for the after-sale service, because of the difficulty in providing its field engineers with the information necessary to maintain cookers of all these different models. In general, field engineers may need to be able to deal with any problem concerning any of the sold cookers, which may include versions previously unknown to them. Producing conventional service manuals and other product documentation for each model variant clearly imposes unacceptable strains on the production cycle, and the resulting volume of documentation will be unmanageable for field engineers. The company periodically issues updated documentation CDs to field engineers as a partial solution, but it has been accepted among its field engineers that more automated and/or intelligent diagnosis support is still needed. This is the broad scope of our research, for which preliminary results have been published (see. e.g. [2], [7], [3], and [8]).

The current system in use for fault diagnosis employs a large after-sale services department consisting of customer call receivers and field engineers. When a customer calls to report a fault, the customer call receiver will try to solve that case through a telephone dialogue. If he/she cannot do so, he/she will record the case in an after-sale services information system as an unsolved case. The system assigns recorded cases to field engineers each day, and field engineers go to the corresponding customers to solve the assigned cases. After solving a case, the field engineer will phone back to the after-sale services department to report the solved case and that case is recorded as completed in the system. All the data about previous cases is stored in the system for quite a long period of time.

It is clear that any system that might make it more likely for a fault to be correctly identified by the customer call receiver, or more rapidly diagnosed by a service engineer, would be of value. In this context, we have designed and implemented a simple case-based diagnosis tool to give the service personnel more intelligent support.

2.2 Diagnosis Process

The diagnosis process of the tool is depicted in Fig. 1. When encountering a new case, the service agent (call receiver or field engineer) will provide a description of the new case according to the customer's report. This description will be matched with the cases in the case base. Some most similar cases will be retrieved, which can be viewed in detail by the service agent to help him/her to identify the fault of the new case. After the new case has been solved, it can be stored into the case base for future diagnosis.

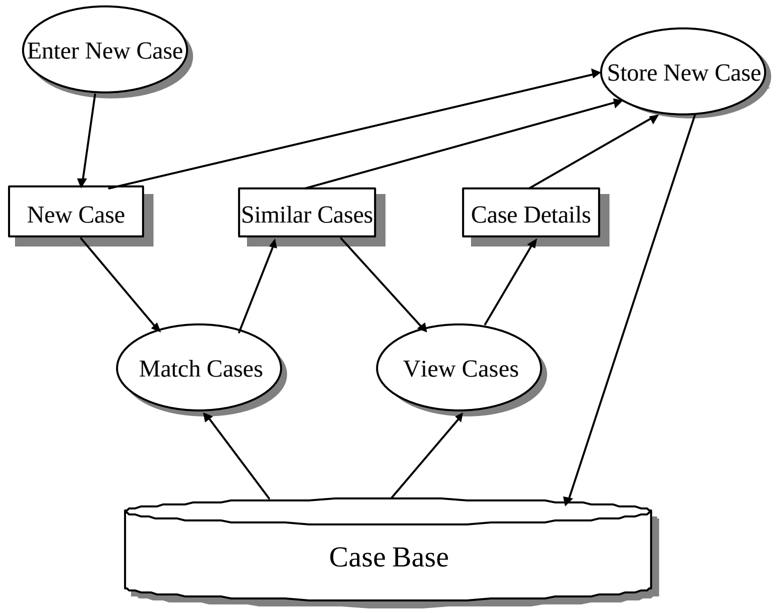

Fig. 1. The diagnosis procedure

2.3 Case Representation

As mentioned above, there is an after-sale services information system for recording maintenance requests of customers and assigning the requests to field engineers. In that system, a case is represented as values in the following attributes (see Table 1).

Table 1. Original Case Attributes

Attribute Name	Data Type
ID	AutoNumber
CallDate	Date/Time
Surname	Text
HouseNo	Text
StreetName	Text
Town	Text
Postcode	Text
PhoneNo	Text
JobNo	Text
Engineer	Text
FaultDescription	Text
FaultCodes1	Number
FaultCodes2	Number
FaultCodes3	Number
FaultCodes4	Number

Among these attributes, most are for identifying the location of the customers and help field engineers to find their customers. As these attributes are irrelevant to diagnosis, we only use five of the above attributes in our diagnosis tool. These attributes are shown in Table 2. An alternative and more sophisticated case structure exploited in the same domain can be found in [8].

Table 2. Case Attributes for Diagnosis

Attribute Name	Data Type
ID	AutoNumber
FaultDescription	Text
FaultCodes1	Number
FaultCodes2	Number
FaultCodes3	Number

The meanings of the four fault codes are as follows. The first fault code is called the *area code*, which denotes the main part of the cooker that the fault is in. For example, the *area code* '6' represents the main oven. The second code is called the *part code*, which denotes the sub-part in the main part. For example, the *part code* '17' represents the door handle. The third code is called the *fault code*, which denotes the actual fault. For example, the *fault code* '55' represents the 'loose wire' fault. The fourth code is called the *action code*, which denotes the action that has been taken to fix the fault. Presently, there are 8 choices for the first code, 194 choices for the second code, 59 choices for the third code, and 26 choices for the fourth code. As our tool is focused on diagnosing the fault, we do not use the *action code* in our tool.

2.4 Case Matching

In this experimental version of our tool, we use a simple case matching strategy. As an engineer can usually find the possible faulty parts in a short time, our tool requires that two similar cases should share the same *area code* and the same *part code*. Another reason of this case matching strategy is that the *area code* and the *part code* are actually representing the location of the fault and therefore cases sharing the same *fault code* under different *area codes* or *part codes* are unlikely to be similar at all. Under this circumstance, the similarity of cases is based only on a matching of the fault descriptions of the cases. As the fault description in a case is in plain English, we simply count the number of matched words between two descriptions as the similarity. The more words are matched, the more similar the two cases are. We require at least one word to be matched, to establish a potential similarity between cases.

3 Diversification in the Tool

3.1 Retrieval Set and Hit Rate

To explain the reason for using diversification techniques in our tool, we first explain the concepts of *retrieval set* and *hit rate*. As the fault descriptions of cases are provided verbally by customers, who may have little knowledge of their cookers, the most similar case identified may not actually exhibit the same fault as the case under diagnosis. To increase the probability that the actual fault will be identified correctly, a set of similar cases is retrieved, rather than just the single most similar case. It is hoped that one of the similar cases may have the same fault as the case under diagnosis. To evaluate the success of the diagnosis, we use the concept '*hit rate*'. The *hit rate* is defined as the number of cases under diagnosis whose faults appear in the faults of their *retrieval set,* divided by the total number of cases under diagnosis. For example, suppose there are 100 cases under diagnosis, and in 80 cases the corresponding retrieval set includes a case that suggests a correct diagnosis of the fault under consideration. Then the *hit rate* is therefore 80%.

Obviously, increasing the size of retrieval sets can usually increase the hit rate. However, as well as the cost of retrieving more cases, a larger retrieval set increases the difficulty in analysing the results to correctly identify the fault, so to general we will aim to restrict the size of the retrieval set. As only those cases that have distinct faults in the retrieval set can actually contribute to an increase in the hit rate, diversification of the retrieval set may also achieve the same effect as increasing the size of the retrieval set. In our tool, we use the following strategy to retrieve only those cases suggesting distinct faults.

3.2 Diversification Strategy

The diversification strategy exploited in our tool is essentially the '*diversification by elimination*' strategy proposed in [4]. However, our strategy is aiming at eliminating cases suggesting the same faults, while the '*diversification by elimination*' strategy is aiming at eliminating *similar cases* (which share the same descriptions with other retrieved cases in most of the attributes).

As discussed in [4], a main limitation of the '*diversification by elimination*' strategy is that it may cause loss in similarity. However, this limitation almost does not exist in our tool, because an eliminated case must have the same values in all the three fault codes (which uniquely identify the actual fault) with a previously retrieved case.

The algorithm for our strategy is depicted in Fig. 2. Supposing the input candidate cases are stored in the variable '*Candidates*' ordered by similarity, and *k* is the maximum size of the *retrieval set*. The output is the variable '*RetrievalSet*' containing at most *k* cases with distinct faults.

Input: Candidates, k
Output: RetrievalSet
Algorithm:
Step 1: RetrievalSet? F
Step 2: C? head(Candidates)
Candidates? tail(Candidates, |Candidates|-1)
if (has-distinct-fault (RetrievalSet, C))
add (RetrievalSet, C)
Step 3: if (|RetrievalSet |?k and |Candidates|?0)
goto Step 2
else exit

Fig. 2. Diversification algorithm

3.3 An Example

Fig. 3 and Fig. 4 show the changes between the normal retrieval set and the diversified retrieval set when diagnosing the same case. The sizes of the two retrieval sets are both seven. Fig. 3 depicts the situation without diversification. The fault '*Inoperative*' appears three times in the retrieval set. Fig. 4 depicts the situation after diversification, where the fault '*Inoperative*' only counts once, and two new faults appear in the retrieval set.

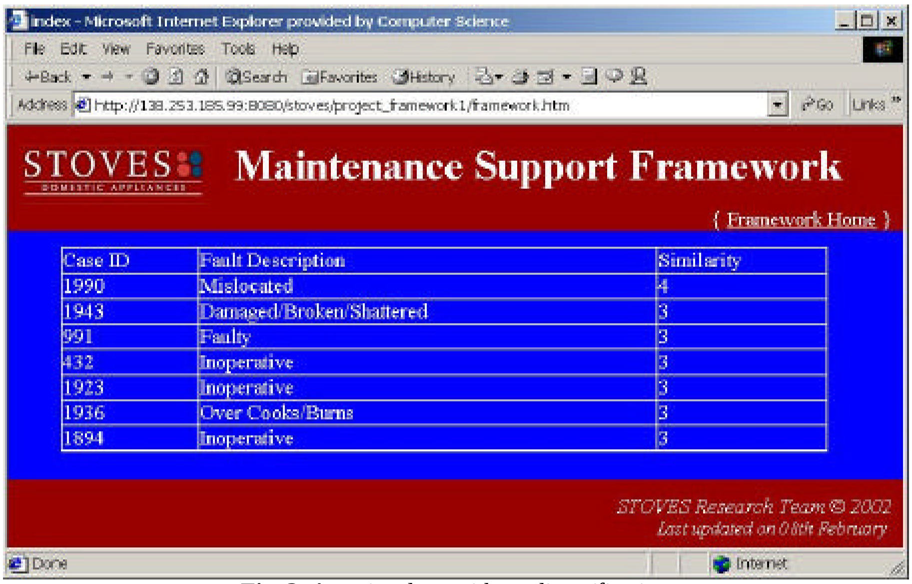

Fig. 3. A retrieval set without diversification

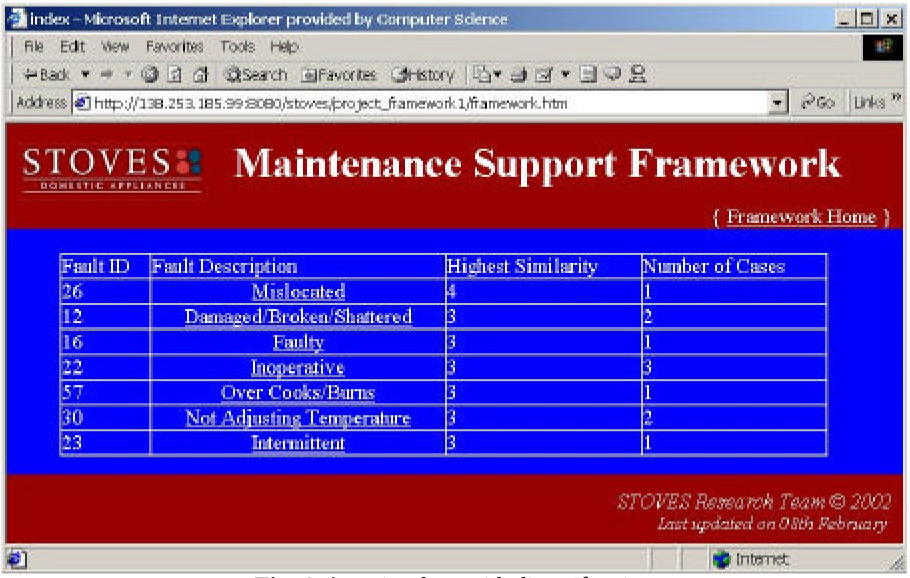

Fig. 4. A retrieval set with diversification

4 Experimental Results

4.1 Experimental Method

To evaluate the benefit of diversification in the tool, we performed some experiments on some real data obtained from the company concerned. We collected 1988 cases recorded in the after-sale services information system during October and November 2001. As the original cases are represented as values in the attributes in Table 1, we extracted only the values in the attributes in Table 2 to form our case base.

We then randomly separated the 1988 cases into a training set containing 1000 cases, used to create the case base, and a test set containing 988 cases. For different retrieval set size k, we recorded both the hit rate of the case-based diagnosis without diversification and the hit rate of the diversified case-based diagnosis. Finally, we represented the relationships between the retrieval set sizes and the two hit rates as a chart containing two lines.

To avoid occasional results, we performed the experiments three times using different random separations. In the following, we report the results of the three experiments.

4.2 Results and Analysis

As the results of the second experiment and the third experiment are similar to those of the first, we report the first experiment in detail, and the other two briefly.

4.2.1 First Experiment

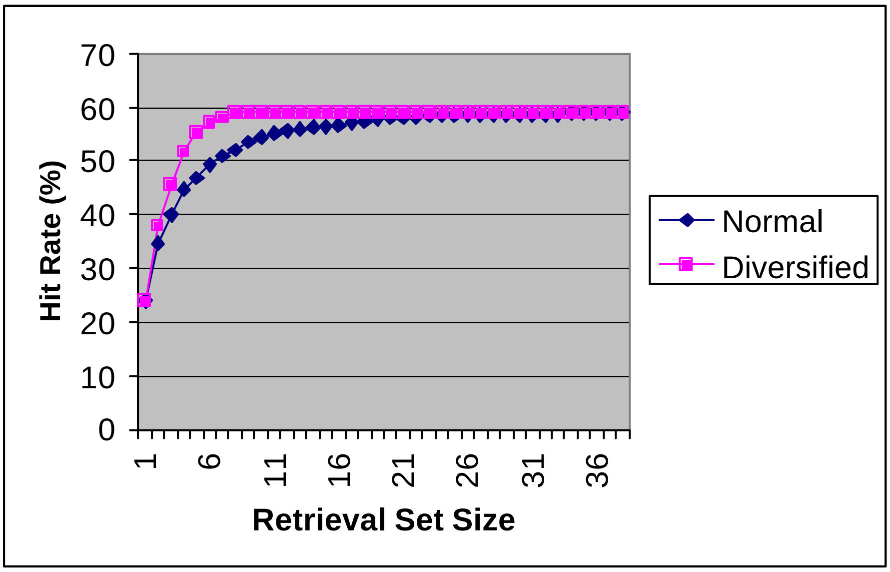

Fig. 5. Results of the first experiment

When the retrieval set size is 1, only the single most similar case is retrieved in both the 'normal' approach and the diversified approach, achieving a hit rate of 23.99%. With the increase of the retrieval set size, both hit rates also increase, but that of the diversified approach increases more rapidly. When the retrieval set size is 5, there is the maximum difference of hit rates between the two approaches – 8.40 percentage points. The diversified approach reaches the highest hit rate (59.11%) when the retrieval set size is 9, while the normal approach reaches the highest hit rate (59.11%) when the retrieval set size is 38. The convergence of the two approaches at this hit rate illustrates the failure of the fault description, in many cases, to provide a good basis for diagnosis. These results show that diversification can help the tool to reach the maximum hit rate when the retrieval set size is still quite small. On average, there is a 7.12 percentage point difference between the two approaches when the retrieval set size is between 4 and 9. The line chart for comparing the hit rates of the two approaches in the first experiment is in Fig. 5.

4.2.2 Second Experiment

The results of the second experiment are similar. When the retrieval set size is 1, both approaches achieve the lowest hit rate of 22.37%. When the retrieval set size is 6, there is the maximum difference of hit rates between the two approaches – 7.59 percentage points. The diversified approach reaches the highest hit rate (60.53%) when retrieval set size is 10, while the normal approach reaches the highest hit rate (60.53%) when retrieval set size is 41. On average, there is a 6.63 percentage point difference between the two approaches when the retrieval set size is between 4 and 9. The line chart for comparing the hit rates of the two approaches in the second experiment is in Fig. 6.

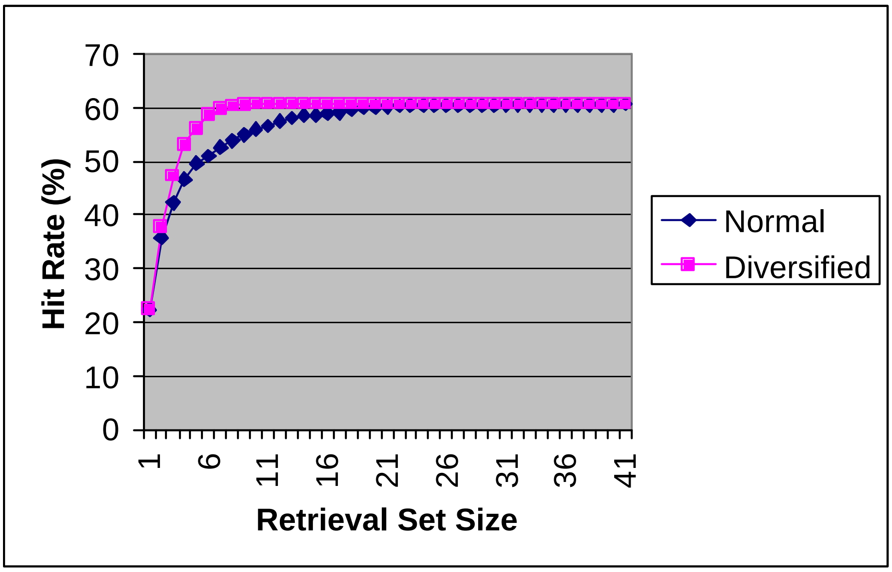

Fig. 6. Results of the second experiment

4.2.3 Third Experiment

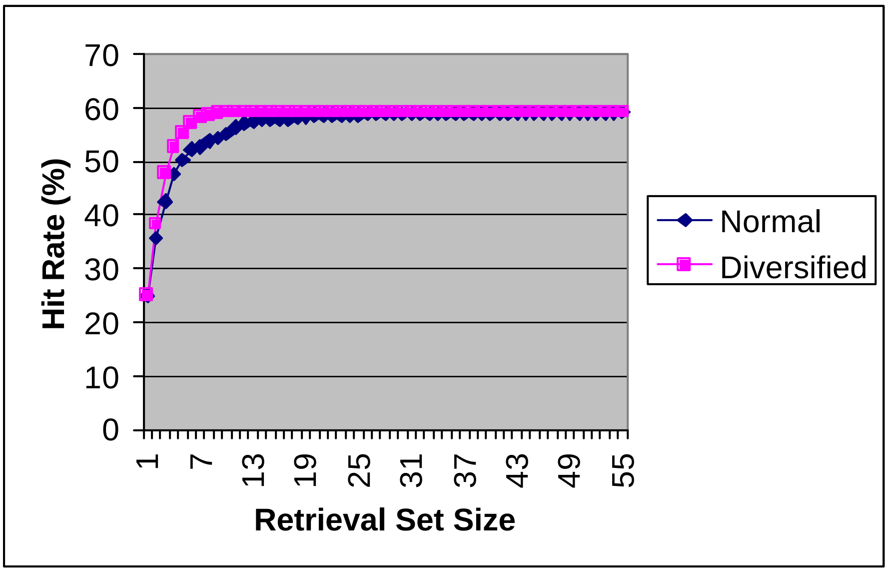

Fig. 7. Results of the third experiment

When the retrieval set size is 1, both approaches achieve the lowest hit rate of 25.00%. When the retrieval set size is 7, there is the maximum difference of hit rates between the two approaches – 5.57 percentage points. The diversified approach reaches the highest hit rate (59.11%) when retrieval set size is 10, while the normal approach reaches the highest hit rate (59.11%) when retrieval set size is 55. On average, there is a 5.06 percentage point difference between the two approaches when the retrieval set size is between 4 and 9. The line chart for comparing the hit rates of the two approaches in the third experiment is in Fig. 7.

4.2.4 Summary

The results of the three experiments are summarised in Table 3.

Table 3. Summary of the experiments

Experiment	1	2	3
Lowest Hit Rate	23.99%	22.37%	25.00%
Highest Hit Rate	59.11%	60.53%	59.11%
Maximum Difference	8.40	7.59	5.57
Average difference (4-9)	7.12	6.63	5.06
Highest set size (diversified approach)	9	10	10
Highest set size (normal approach)	38	41	55

5 Limitations

Our main concern of this study is the poor quality of the case data. These data are collected from an information system exploited in Stoves for managing the maintenance process, and are not planned particularly for case matching. First, our case matching is mainly based on matching free texts. This may make the basis of the case-based diagnosis very weak. Secondly, the text descriptions of the cases are provided by individual customers who know little about the cookers. This may result in much noise in the text descriptions. Due to the above two factors, our similarity metric might have been much distorted. Actually, there is already some evidence that our tool cannot find the real faults very effectively. This is also one reason that we resort to the diversified approach. The question is to what extent the poor quality can affect our main conclusion. Although there is no direct evidence that the poor quality can definitely lead to a wrong conclusion, it may increase the probability that our conclusion is merely an occasional conclusion.

Another concern is about the confidence of our experiments. First, we only performed three experiments. So, the conclusion drawn from three experiments may not have a high confidence level. Secondly, some important parameters in our experiments are not carefully tuned. Is the number of total cases enough or should we acquire more cases from our partner for the experiments? Is the training set large enough or not?

Based on the above two concerns, we think our conclusion should only be tentative and still requires further evaluation.

6 Conclusion

In this paper, we have reported an experiment in applying a diversification technique to a case-based diagnosis tool in a product maintenance domain. The aim of our study is to evaluate whether and to what extent applying the diversification technique in the tool is advantageous over not applying the technique. Our experimental results show that, in the real-life domain we have investigated, the diversified approach is much more effective than the un-diversified approach in successfully including the real fault within a relatively small retrieval set. This result arises, in part, because of the relatively poor quality of the data used for matching cases. In this context, there is a relatively high probability that the correct fault will not appear within a small retrieval set, and, without diversification, it sometimes requires a very large retrieval set to produce the most successful match. We have shown that in this case, use of diversification can be very effective in identifying the best matches without the need for an unmanageably large retrieval set. Therefore, we conclude that diversification may be an effective method to increase the *hit rate* while keeping a rather small retrieval set size in a noisy diagnosis domain.

Acknowledgements

The research described in this paper is supported by the Department of Trade and Industry under the Foresight ‘Link’ programme. We wish to express our thanks to our partners in the project, Stoves PLC and NA Software Ltd, and in particular to Colin Johnson, Prof Mike Delves, and the Project Monitor, Stan Price.

References

1. Bradley, K., Smyth, B.: Improving Recommendation Diversity. In: Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland (2001) 85-94
2. Coenen, F., Leng, P., Weaver, R., Zhang, W.: Integrated online support for field service engineers in a flexible manufacturing context. In: Applications and Innovations in Intelligent Systems VIII (Proc ES2000 Conference, Cambridge), eds A Macintosh, M Moulton and F Coenen, Springer, London, (2000) 141-152
3. Coenen, F., Leng, P., Zhang. L.: Flexible Field Service Support using Multiple Diagnostic Tools. In Proceedings of 5th IEEE International Conference on Intelligent Engineering Systems, (2001) 225-229
4. McSherry, D.: Increasing Recommendation Diversity Without Loss of Similarity. In: Proceedings of the Sixth UK CBR Workshop, 10 December (2001) 23-31
5. Smyth, B., Cotter, P.: A Personalised TV Listing Service for the Digital TV Age. Knowledge-Based Systems, 13 (2000) 53-59.
6. Smyth, B., McClave, P.: Similarity vs. Diversity. In: Aha, D.W., Watson, I. (eds) Case-Based Reasoning Research and Development. LNAI, Vol. 2080. Springer-Verlag, Berlin Heidelberg (2001) 347-361

7. Zhang, W., Coenen, F., Leng, P.,: On-Line Support for Field Service Engineers in a Flexible Manufacturing Environment: the Stoves Project. In: Proceedings of IeC '2000 Conference, Manchester (2000) 31-40
8. Zhang, L., Coenen, F., Leng, P.,: A Case Based Diagnostic Tool in a Flexible Manufacturing Context. In: Proceedings of the Sixth UK CBR Workshop, 10 December (2001) 61-69