Can Domain Knowledge Help 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. The quality of case data can be an important factor for a Case Based Reasoning (CBR) system. In this paper, we report an experimental study in using domain knowledge to help case-based diagnosis in a product maintenance context. For the problem we are facing, cases are based on the descriptions of customers who have little knowledge of their products. As each case can be associated with some domain knowledge provided by domain experts, we try to use the domain knowledge to assist case matching. Our experimental results show that the domain knowledge can significantly improve the performance of the case-based diagnosis.

Keywords. Diagnosis, Hybrid CBR, Product maintenance

1 Introduction

Case Based Reasoning (CBR) is a multi-disciplinary subject that focuses on the reuse of experiences [1]. In particular, CBR has been applied to solve diagnosis problems (see e.g. [2] and [5]). For a case-based diagnosis system, the quality of case data is usually a key factor of the success of the system. In our research, we are facing a diagnosis problem in a product maintenance domain, in which cases are mainly described by customers having very little knowledge of their products. Therefore, it cannot achieve a good performance by simply comparing the text description of the new case with those of the previous cases.

Fortunately, the company concerned uses a coding system to identify all the possible faults associated with the cases. Under this coding system, a fault contains both the location and the nature of the fault. As the location of a fault is usually quite easy to know for a domain expert, we can view the location information as the domain knowledge associated with each case. Therefore, it might be a good idea to involve the location information into case matching. Similar ideas have already investigated in text classification (see e.g. [6] and [7]).

In this paper, we report an experimental study of using part and all the location information as domain knowledge in case matching. According to our preliminary results, using more domain knowledge can achieve higher hit rates in this domain.

The remainder of this paper is organised as follows. Section 2 provides a general description of the problem we are facing. Section 3 reports the setup of our experiments. Section 4 reports the experimental results and the corresponding analysis. Section 5 concludes this paper.

2 Background

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. [3], [8], [4], [9], and [10]).

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 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 [9].

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 examp le, 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.

3 The Experiments

3.1 Three Case Matching Strategies

From the above case representation, the original diagnosis problem is as follows. Given a text description of a new fault, our diagnosis system should try to find the three fault codes via matching the text description against previous cases. As the text descriptions are provided by customers who have very little knowledge about cookers, it is predictable that any method would not solve the problem very effectively. As the rough location of a fault is fairly easy for an engineer to find, a method can assume that some or even all location knowledge can be provided by the engineer.

```
Input: CaseBase, NewCase, k
Output: RetrievalSet
Strategy 1:
    Step 1: RetrievalSet?  F
    Step 2: for each i in CaseBase
                If (similarity(i.description, NewCase.description)>0)
                    Add(RetrievalSet, i)
    Step 3: sort RetrievalSet according to similarity
    Step 4: delete cases other than most similar k cases in RetrievalSet

Strategy 2:
    Step 1: RetrievalSet?  F
    Step 2: for each i in CaseBase
                If (i.area=NewCase.area and
                   similarity(i.description, NewCase.description)>0)
                    Add(RetrievalSet, i)
    Step 3: sort RetrievalSet according to similarity
    Step 4: delete cases other than most similar k cases in RetrievalSet

Strategy 3:
    Step 1: RetrievalSet?  F
    Step 2: for each i in CaseBase
                If (i.area=NewCase.area and and i.part=NewCase.part and
                   similarity(i.description, NewCase.description)>0)
                    Add(RetrievalSet, i)
    Step 3: sort RetrievalSet according to similarity
    Step 4: delete cases other than most similar k cases in RetrievalSet
```

Fig. 1. Three case matching strategies

In our experiments, we evaluate whether the location knowledge can benefit the diagnosis. The first strategy examined is aiming at the original problem – just matching the text descriptions against previous descriptions. The second strategy is to assume that the correct area code can be provided by the engineer and thus can be used in the matching process. In this strategy, only the cases that share the same area code with the new case are matched against the new case. The third strategy is to assume that both the correct area code and the correct part code can be provided by the engineer and thus can be used in the matching process. In this strategy, only the cases that share both the same area code and the same part code with the new case are matched against the new case. The three strategies are illustrated in Fig 1.

3.2 Retrieval Set and Hit Rate

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, in general we will also aim to restrict the size of the retrieval set. In our experiments, we record the hit rates of the three strategies under various retrieval set sizes.

3.3 Experimental Process

To evaluate the performance of the above three case matching strategies, 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 rates of the three case matching methods. Finally, we represented the relationships between the retrieval set sizes and the three hit rates as a chart containing three lines. To avoid occasional results, we performed the experiments three times using different random separations.

4 Experimental Results

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.

4.1 First Experiment

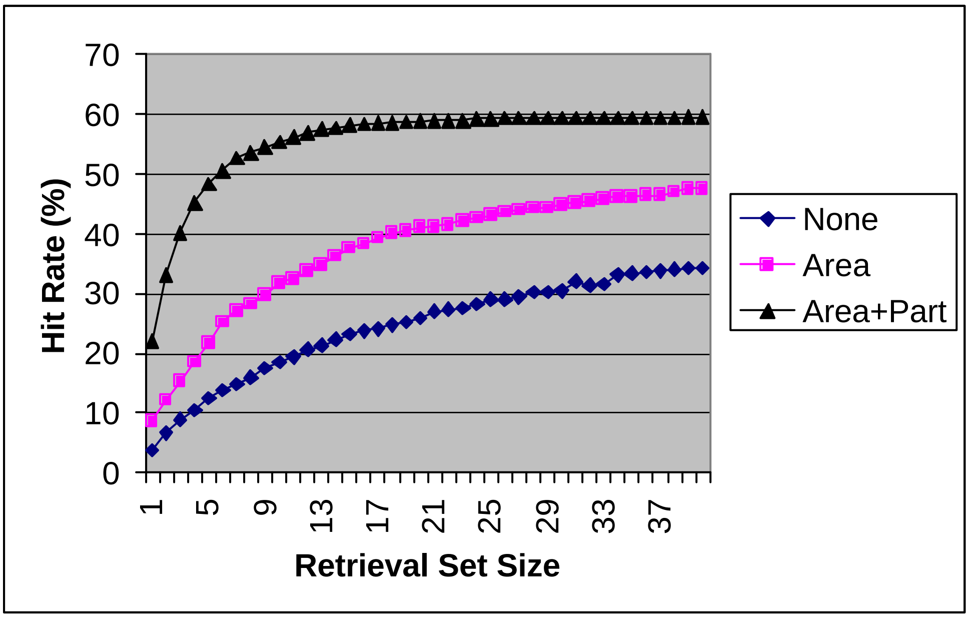

Fig. 2. Results of the first experiment

The line chart for comparing the hit rates of the three strategies in the first experiment is in Fig. 2. In general, the hit rates of all the three strategies will increase with the increase of the retrieval set sizes. Whatever the retrieval set size is, the third strategy (using both the area code and the part code as domain knowledge) is always significantly higher than the second strategy (using only area code as domain knowledge) and the first (not using any domain knowledge). When the retrieval set size is 4, there is the maximum difference of hit rates between the third strategy and the second strategy – 26.72 percentage points. When the retrieval set size is 7, there is the maximum difference of hit rates between the third strategy and the first strategy – 37.75 percentage points. On average, there is a 25.34 percentage point difference between the third strategy and the second strategy, and a 35.94 percentage point difference between the third strategy and the first strategy, when the retrieval set size is between 3 and 10. From this, we using domain knowledge for this problem can really increase the hit rates.

From Fig. 2, we can see that highest hit rate of the third strategy is only around 60%. This indicates that even using the area code and the part code as domain knowledge, our case data still cannot be a good basis for diagnosis. However, the curve of the third strategy in Fig. 2 can indicate another merit of using domain knowledge. By using domain knowledge, the third strategy can nearly reach the highest hit rate when the retrieval set size is still manageable. In this experiment, when the retrieval set size is 13, the hit rate of the third strategy can reach 57.49%, which is only lower than the highest rate by 2.02 percentage points.

4.2 Second Experiment

The results of the second experiment are similar. The line chart for comparing the hit rates of the two approaches in the second experiment is in Fig. 3. The third strategy still has higher hit rates than the other two strategies. When the retrieval set size is 5, there is the maximum difference of hit rates between the third strategy and the second strategy – 24.60 percentage points. When the retrieval set size is 6, there is the maximum difference of hit rates between the third strategy and the first strategy – 37.55 percentage points. On average, there is a 23.96 percentage point difference between the third strategy and the second strategy, and a 36.32 percentage point difference between the third strategy and the first strategy, when the retrieval set size is between 3 and 10.

The highest hit rate of the third strategy is still around 60%. When the retrieval set size is 13, the hit rate of the third strategy can reach 57.89%, which is only lower than the highest rate by 2.53 percentage points.

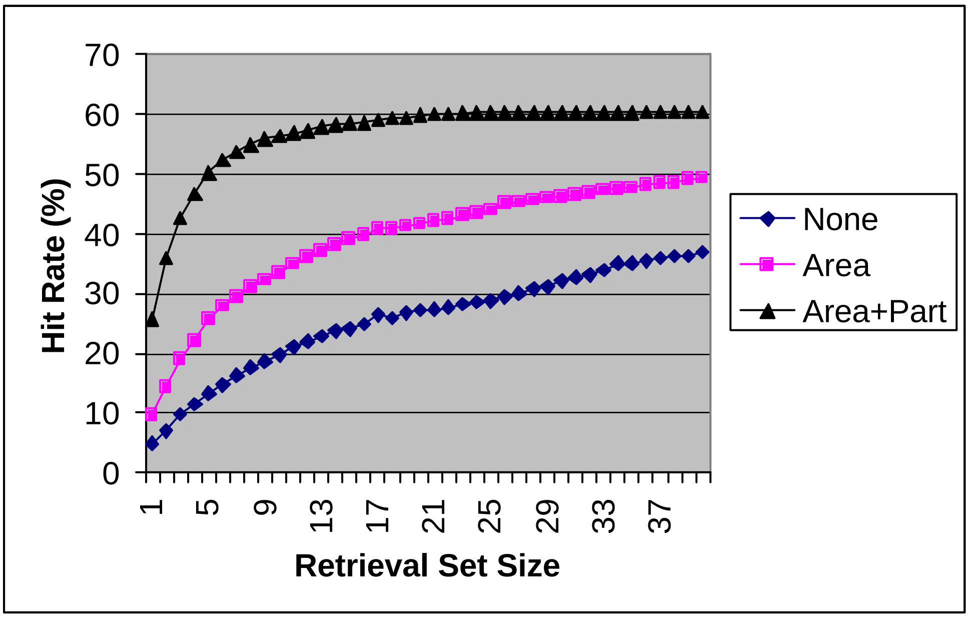

Fig. 3. Results of the second experiment

4.3 Third Experiment

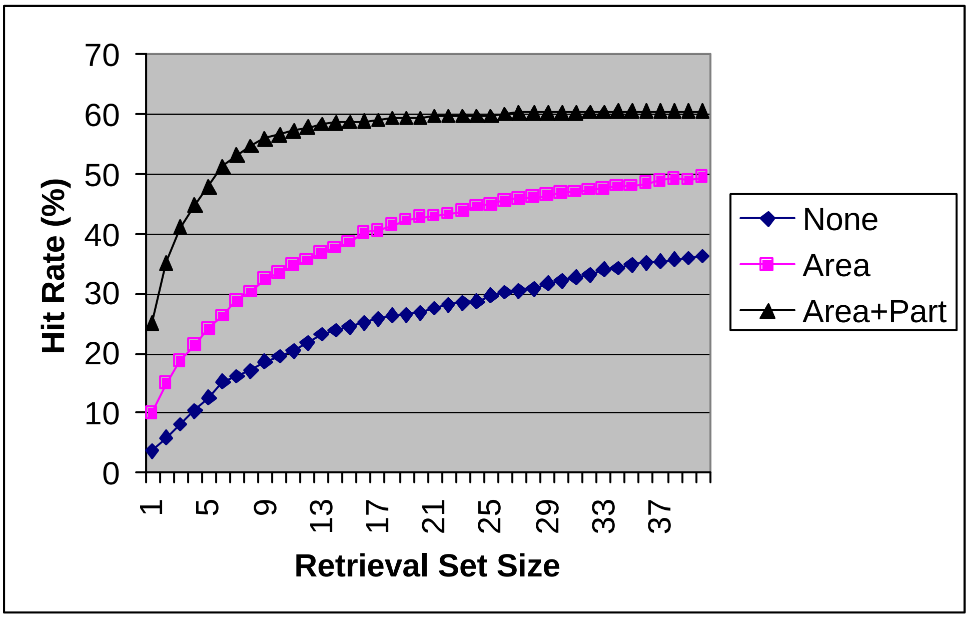

Fig. 4. Results of the third experiment

The line chart for comparing the hit rates of the two approaches in the second experiment is in Fig. 4. The third strategy still has higher hit rates than the other two strategies. When the retrieval set size is 6, there is the maximum difference of hit rates between the third strategy and the second strategy – 25.00 percentage points. When the retrieval set size is 8, there is the maximum difference of hit rates between the third strategy and the first strategy – 37.55 percentage points. On average, there is a 23.70 percentage point difference between the third strategy and the second strategy, and a 35.89 percentage point difference between the third strategy and the first strategy, when the retrieval set size is between 3 and 10.

The highest hit rate of the third strategy is still around 60%. When the retrieval set size is 13, the hit rate of the third strategy can reach 58.30%, which is only lower than the highest rate by 2.23 percentage points.

4.4 Summary

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

Table 3. Summary of the experiments

Experiment	1	2	3
Maximum Difference between Strategy 3 and Strategy 2 (Retrieval Set Size)	26.72 (4)	24.60 (5)	25.00 (6)
Maximum Difference between Strategy 3 and Strategy 1 (Retrieval Set Size)	37.75 (7)	37.55 (6)	37.55 (8)
Average Difference between Strategy 3 and Strategy 2 (3-10)	25.34	23.96	23.70
Average Difference between Strategy 3 and Strategy 1 (3-10)	35.94	36.32	35.89
Difference with Highest (Strategy 3 when the Retrieval Set Size is 13)	2.02	2.53	2.23

5 Conclusion

In this paper, we have reported an experimental study of using domain knowledge provided by domain experts to improve diagnosis based on poor case data in a product maintenance context. The aim of our study is to evaluate whether and to what extent the domain knowledge can bring benefits to the diagnosis process. Our experimental results show that, in the real-life domain we have investigated, the more correct domain knowledge is applied in case matching, the higher hit rates will be achieved. Another finding is that when there is more domain knowledge applied in case matching, it is more likely for the approach to achieve a hit rate approximate to the highest hit rate while the retrieval set size is still fairly small.

Our main concern is the relatively poor quality of the case data. Although this is also the reason why we want to use domain knowledge to increase the hit rates, this may make our conclusions too straightforward, because when the data quality is low, any accurate knowledge might help.

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. Aha, D. W.: The Omnipresence of Case-Based Reasoning in Science and Application. Knowledge-Based Systems, 11(5-6), (1998) 261-273
2. Auriol, E., Crowder, R. M., McKendrick, R., Rowe, R.: Integrating Case-Based Reasoning and Hypermedia Documentation: An Application for the Diagnosis of a Welding Robot at Odense Steel Shipyard. Engineering Applications of Artificial Intelligence, Vol. 12, (1999) 691-703

3. 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
4. Coenen, F., Leng, P., Zhang. L.: Flexible Field Service Support using Multiple Diagnostic Tools. In Proceedings of 5th IEEE International Conference on Intelligent Engineering Sy stems, (2001) 225-229
5. Varma, A., Roddy, N.: ICARUS: Design and Deployment of a Case-Based Reasoning System for Locomotive Diagnosis. Engineering Applications of Artificial Intelligence, Vol. 12, (1999) 681-690
6. Zelikovitz, S., Hirsh, H.: Improving Short Text Classification Using Unlabeled Background Knowledge to Assess Document Similarity. In: Proceedings of the Seventeenth International Conference on Machine Learning, (2000) 1183–1190
7. Zelikovitz, S., Hirsh, H.: UsingLSI for Text Classification in the Presence of Background Text. In: Proceedings of the Tenth Conference for Information and Knowledge Management (2001) 113-118
8. 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
9. 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
10. Zhang, L., Coenen, F., Leng, P.: An Experimental Study of Increasing Diversity for Case-Based Diagnosis. In: Proceedings of 6th European Conference on Case Based Reasoning (ECCBR), 4-7 September (2002) Advances in Case-Based Reasoning, LNAI 2416, 448-459