Using Domain Knowledge to Boost Case-Based Diagnosis: An Experimental Study in a Domain with Very Poor Data Quality

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 our research, we are facing a diagnostic problem in a product maintenance domain, where a large volume of low quality case data is collected. In this paper, we report an experimental study of whether using domain knowledge can improve the performance of either non-diversified or diversified retrieval in this domain. As each case can be associated with some domain knowledge that can be easily obtained in this domain, we try to use it to assist case matching. Our experimental results show that the domain knowledge can significantly improve the performance of both the non-diversified and the diversified approaches. Furthermore, using the diversified approach seems to be of greater value when also using domain knowledge.

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 [7]). An obvious advantage of case-based diagnostic systems is that these systems can be built without detailed information about the target domains. In such a case-based diagnosis system, the quality of case data is usually a key factor in the success of the system. However, in many domains there are already a large number of accumulated low quality cases. It therefore is useful to find a way to improve the data quality and thus make them usable for diagnostic purposes.

In our research, we are facing a diagnostic 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.

In this paper, we report our study of using domain knowledge to improve the data quality. Using domain knowledge for text classification has been studied in [8] and [9]. A preliminary report of this approach can be found in [13]. In our study, we use the location information generated for each case by human semi-experts as the domain knowledge, which will be used in the matching process. Our experimental results show that using domain knowledge can substantially increase the overall performance of the diagnostic system.

In a previous study, we found that diversifying retrieved cases (see e.g. [5], [6], [3], and [4]) can enhance the performance in this domain [12]. In this paper, we also use domain knowledge for the diversified approach. Our experimental results show that using domain knowledge can also improve the diversified approach. We also examine the impact of using domain knowledge on diversification. We find that diversification seems more effective when more domain knowledge is involved in the matching process.

The remainder of this paper is organised as follows. Section 2 provides a general description of the background of the domain. Section 3 presents the method of using domain knowledge in our domain. Section 4 describes the setting of our experiment. Section 5 presents the main experimental results. Section 6 concludes this paper.

2 The Stoves Project

2.1 The Diagnostic Problem

The diagnostic 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 current system used in Stoves employs a large after-sale services department consisting of customer call receivers and field engineers. The product maintenance procedure is depicted in Fig. 1. 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.

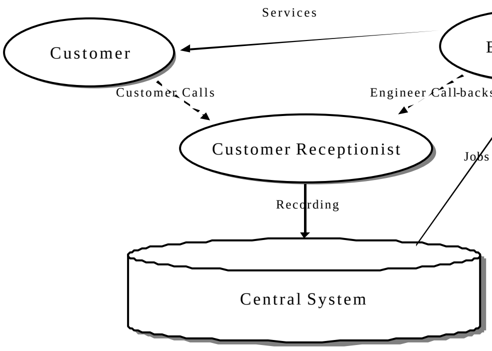


Fig. 1. Stoves' product maintenance procedure

2.2 Case-Based Diagnosis for Stoves

The Stoves Project [10] is a joint project supported by the DTI under the Foresight 'Link' programme, which is carried out in collaboration with Stoves PLC and some other industrial partners. The basic objective of the Stoves Project is to help Stoves PLC to decrease their maintenance costs. As a large maintenance department is managed in Stoves, a little progress may mean saving quite a large amount of money. Ideally our approach should also be generic and easy to be adapted for other manufacturers. To achieve this objective, we exploit a case-based approach [11]. The benefits for us are mainly two-fold. Firstly, none of the researchers really understand the mechanism of cookers, and in a case-based approach we can avoid knowing this in detail. This also makes the approach more generic than would be a method tailored to the product domain. Secondly, this approach allows us to take advantage of all the data recorded in the current system in Stoves. Typically, there will be more than 500 cases in one month.

However, the quality of the case data is a major concern. In fact, all the case descriptions are basically the telephone reports from the customers. As customers typically know very little about cookers, their reports are usually very imprecise. For this reason, our initial diagnostic system failed to achieve a satisfactory performance in some preliminary experiments.

3 Using Domain Knowledge

In Stoves' system, cases are mainly represented as a table in the database. The attributes of the table are listed in Table 1.

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

Table 1. Original Case Attributes

A simple coding system is used for recording faults and corresponding actions. A fault and its corresponding action are recorded as four codes. 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. The four codes are referred to as the four fault codes in Table 1.

From the above case representation, the original diagnostic problem is as follows. Given a text description of a new fault, the diagnosis system should try to find the three fault codes and the action code 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 understandable that any method would not give a good performance. However, typically a customer call receiver can easily find the rough location of the reported fault, and record the location with the case description. Therefore, we think that using the location information provided by customer call receivers as domain knowledge in case matching may improve the quality of cases.

To evaluate whether and to what extent this case matching strategy using domain knowledge is beneficial, we performed an experimental study. Among the attributes in Table 1, 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 actually do not use them in our study. As we are interested in diagnosis, we also ignore the *action code*. The attributes used in our study are listed in Table 2.

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

Table 2. Case Attributes in the Study

4 The Experiments

4.1 Experimental Subject: Three Case Matching Strategies

In our experiments, we evaluated three strategies using different amounts of location knowledge in the matching process. 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 customer call receiver 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 customer call receiver 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 2.

The three strategies share the same similarity measure when determining the similarity between the description of a case in the case base and that of the new case. We exploit a simple method for matching two texts: After eliminating some stop-words (such as articles) in the two texts, we count the number of matched words as the similarity.

4.2 Experimental Objectives

The basic objective was to evaluate the performance of the above three strategies. What we hoped was to see the strategy using most domain knowledge to win the contest, although this seems quite natural. As in our previous study we found that diversification may increase the performance of diagnosis in this domain, we also wished to know what the performance of the three strategies would be when the retrieval set is diversified. We expected the strategy using most domain knowledge to also win in this context. Another interesting point is how the idea of using domain knowledge and the idea of diversification will interplay. Hopefully, they can be bound together to form the best solution.

```
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. 2. Three case matching strategies

4.3 Evaluation Criterion: 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. Therefore, a well-trained user can analyse the retrieval set to find the fault. To evaluate the success of the retrieval, we use the concept of '*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. If, for example, 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 it is ideal to have a high hit rate when the retrieval set size is still small. In our experiments, we record the hit rates of the three strategies under various retrieval set sizes.

4.4 Experimental Process

To evaluate the performance of the above three case matching strategies, we performed some experiments on some real data obtained from Stoves. We collected 1988 cases recorded in the after-sale services information system during a period in 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 values of the retrieval set size k, we recorded the hit rates of the three case matching methods with and without diversification. To avoid occasional results, we performed the experiments three times using different random separations.

5 Experimental Results

5.1 Using Domain Knowledge without Diversification

The results of the first experiment on using domain knowledge without diversification are depicted in Fig. 3. 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, it is clear that using domain knowledge for this problem can significantly increase the hit rates.

We can see that highest hit rate of the third strategy is only around 60%, so even using the area code and the part code as domain knowledge gives only moderate success in diagnosis using this poor-quality case data. However, the curve of the third strategy in this figure indicates another merit of using domain knowledge. By using domain knowledge, the highest hit rate is approached 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 reaches 57.49%, only 2.02 percentage points less than the maximum, and a hit rate of over 50% is achieved with only 6 cases.

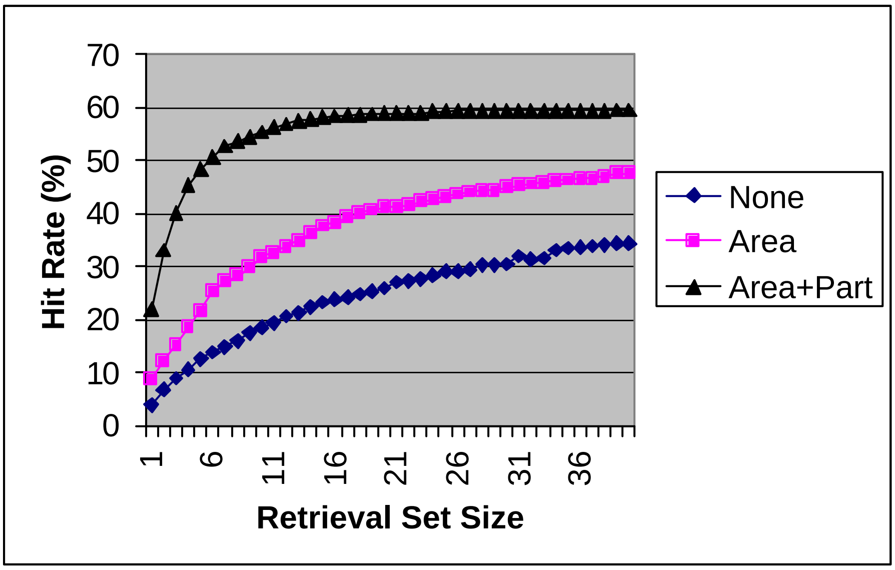


Fig. 3. Using domain knowledge without diversification (experiment 1)

Similar results were obtained in the second and third experiments, depicted in Fig. 4 and Fig. 5. The results of the three experiments are summarised in Table 3.

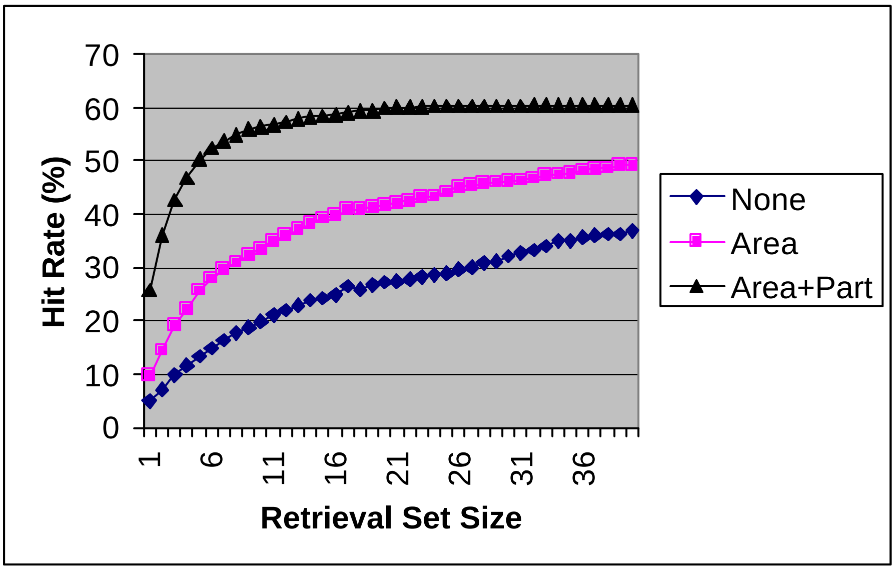


Fig. 4. Using domain knowledge without diversification (experiment 2)

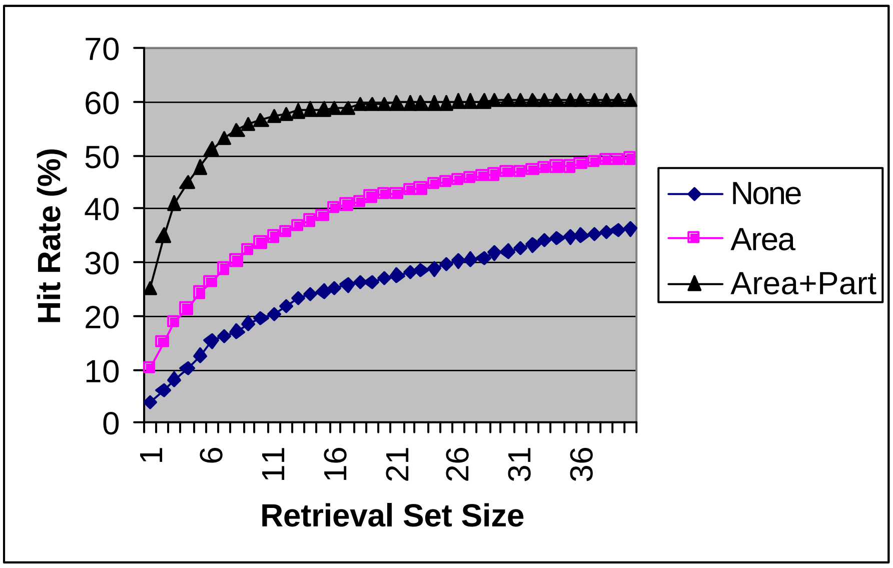


Fig. 5. Using domain knowledge without diversification (experiment 3)

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

Table 3. Summary of experiments on using domain knowledge without diversification

5.2 Using Domain Knowledge with Diversification

The second set of experiments we conducted examined the effect of using this domain knowledge together with a strategy of ‘diversification by elimination’ [4]. The strategy eliminates from the retrieval set cases that suggest the same fault, retaining at most k cases with distinct fault codes. The results of the first experiment using diversification are depicted in Fig. 6. 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 – 32.79 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 – 43.42 percentage points. On average, there is a 29.07 percentage point difference between the third strategy and the second strategy, and a 41.46 percentage point difference between the third strategy and the first strategy, when the retrieval set size is between 3 and 10. From this, we see that using domain knowledge can also increase the hit rates for the diversified approach.

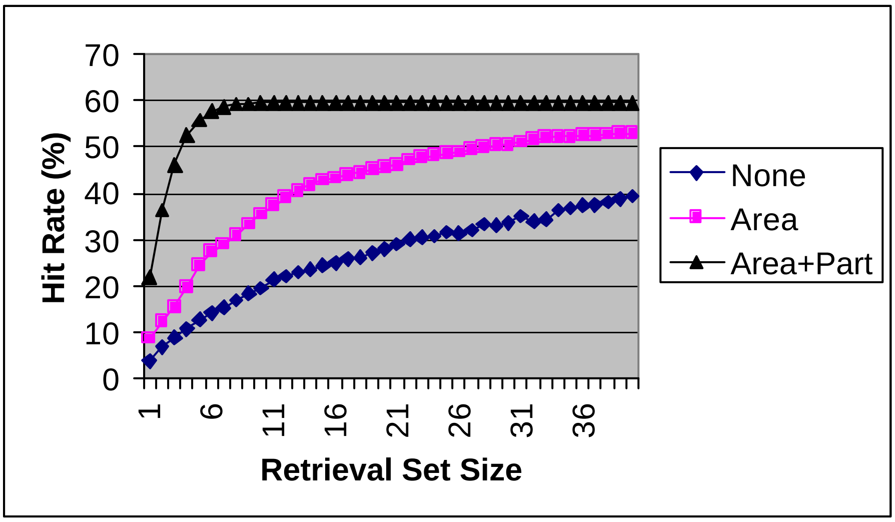


Fig. 6. Using domain knowledge with diversification (experiment 1)

As with the results in the experiments on the non-diversified approach, the curve of the third strategy in this figure indicates that this 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 7, the hit rate of the third strategy reaches 58.60%, almost the maximum. Similar results were obtained in the second and third experiments, depicted in Fig. 7 and Fig. 8. The results of the three experiments are summarised in Table 4.

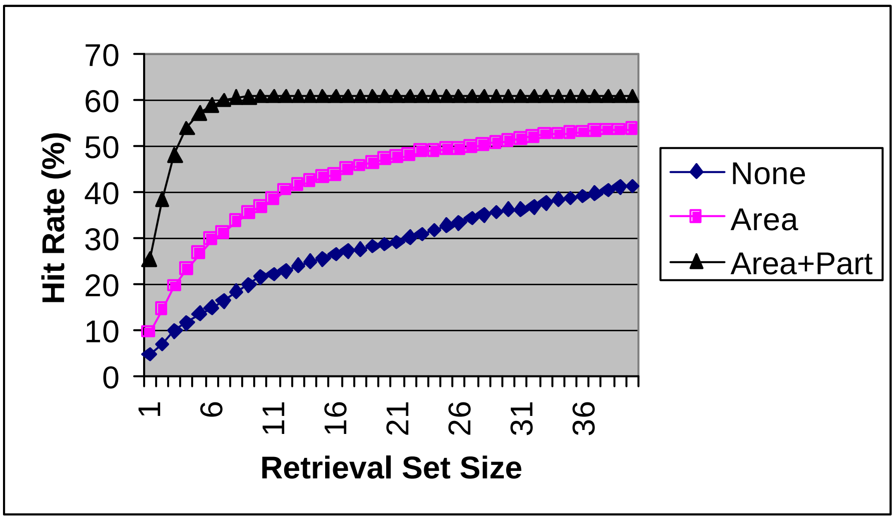


Fig. 7. Using domain knowledge with diversification (experiment 2)

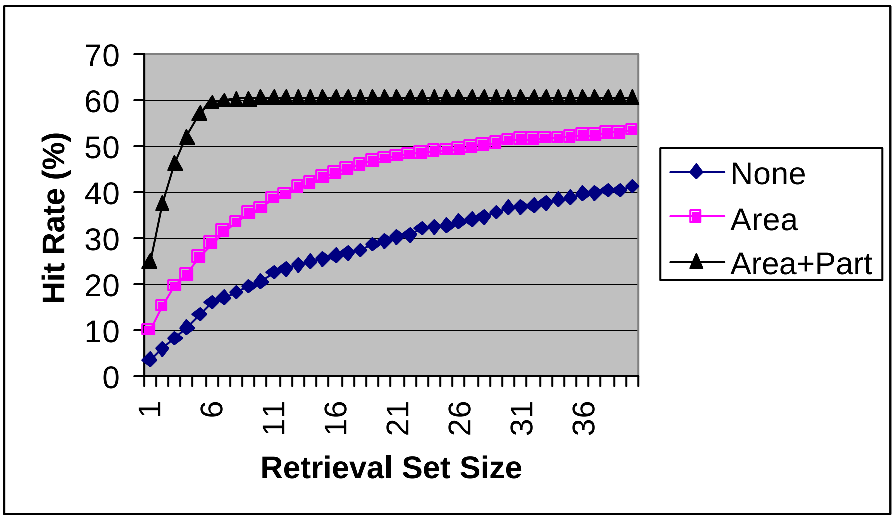


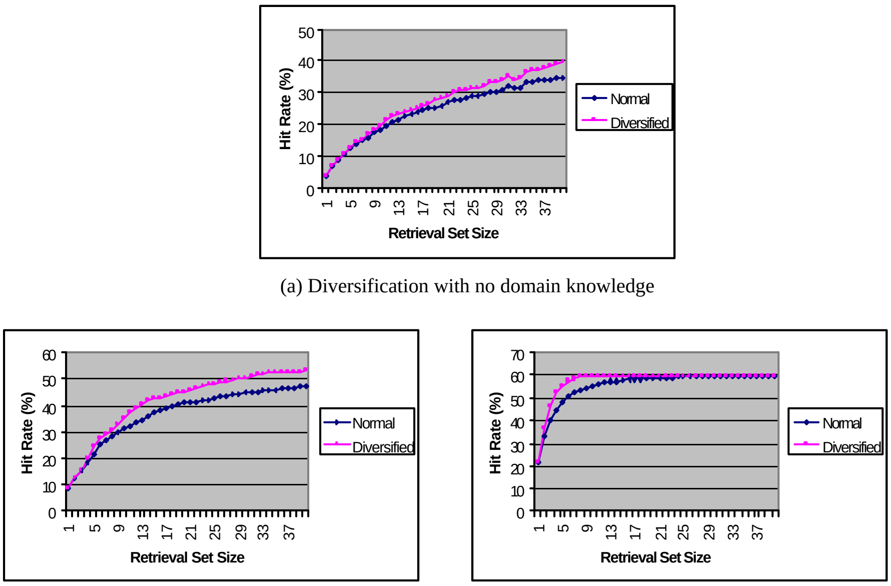
Fig. 8. Using domain knowledge with diversification (experiment 3)

Experiment	1	2	3
Maximum Difference between Strategy 3 and Strategy 2 (Retrieval Set Size)	32.79 (4)	30.36 (5)	31.17 (5)
Maximum Difference between Strategy 3 and Strategy 1 (Retrieval Set Size)	43.42 (6)	43.72 (6)	43.52 (5)
Average Difference between Strategy 3 and Strategy 2 (3-10)	29.07	27.94	27.81
Average Difference between Strategy 3 and Strategy 1 (3-10)	41.46	41.59	41.40
Difference with Highest (Strategy 3 when the Retrieval Set Size is 7)	0.91	0.91	0.61

Table 4. Summary of experiments on using domain knowledge with diversification

5.3 Impact of Domain Knowledge on Diversification

Finally, we examined the impact of domain knowledge on the performance of diversification by reorganising the results shown in Fig. 3 – Fig. 8.



(a) Diversification with no domain knowledge

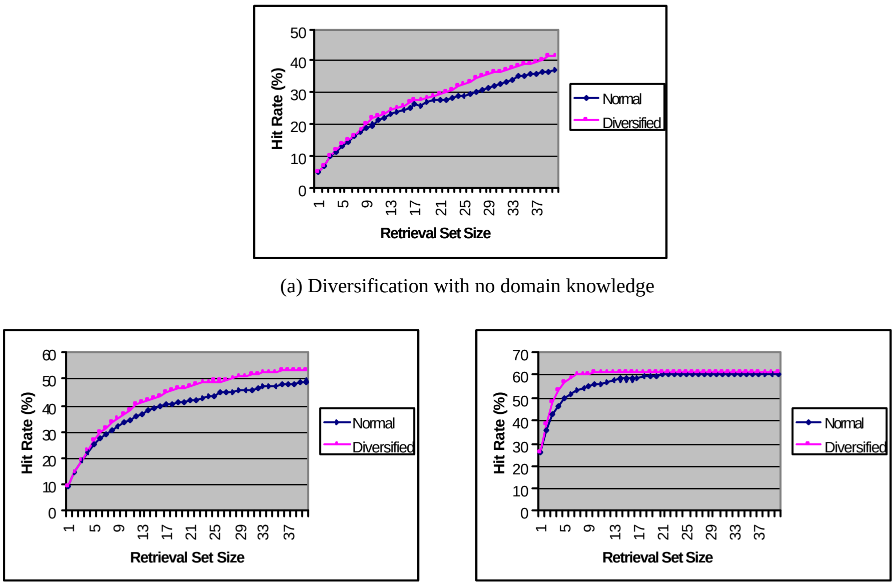
(b) Diversification with area knowledge

(c) Diversification with area and part knowledge

Fig. 9. Impact of domain knowledge on diversification (experiment 1)

Fig. 9 depicts three line charts reformed from Fig. 3 and Fig. 6 based on the first experiment. In Fig. 9(a) where no domain knowledge is used in case matching, the diversified approach and the normal approach cannot be separated when the retrieval set size is between 1 and 10, and the difference then rises steadily as the set becomes larger. However, in Fig. 9(b) where the area knowledge is used, the two approaches are separable when the retrieval set size is larger than 4, and reaches a maximum when the retrieval set size is 33. In Fig. 9(c) where both area knowledge and part knowledge are used in case matching, the diversified approach and the normal approach can be separated when the retrieval set size is between 2 and 16. The difference between the two approaches is greatest, 7.48 percentage points, when the retrieval set size is only 5.

From the above results, it seems that the more domain knowledge is used, the more the advantage of the diversified approach is over the non-diversified approach. Firstly, the more domain knowledge is used, the more likely it is that the two approaches are separable when the retrieval set size is small. Secondly, the more domain knowledge is used, the smaller the retrieval set size is, when the difference between the two approaches is maximised. Finally, the more domain knowledge is used, the bigger is the maximum difference between the approaches. Similarly, we can find the same impacts in the second experiment and the third experiment, whose results are depicted in Fig. 10 and Fig. 11.

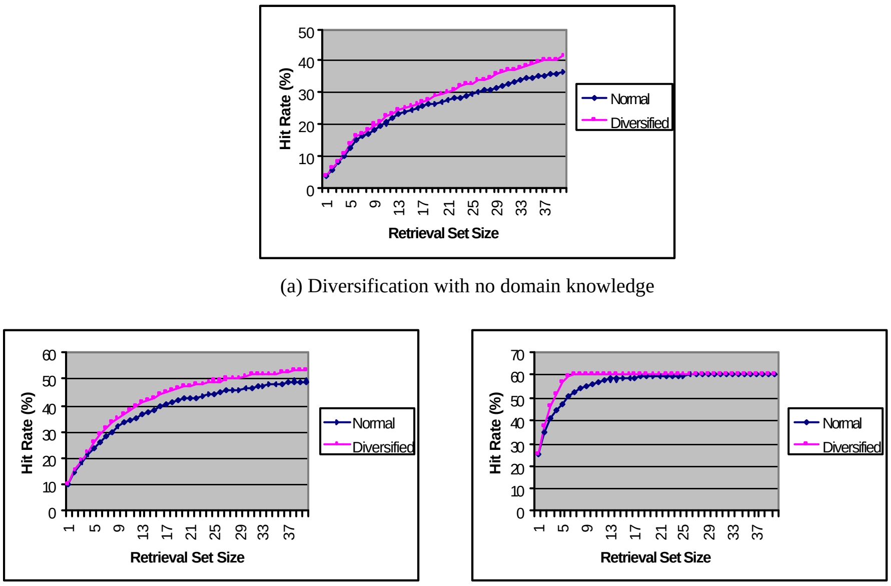


(a) Diversification with no domain knowledge

(b) Diversification with area knowledge

(c) Diversification with area and part knowledge

Fig. 10. Impact of domain knowledge on diversification (experiment 2)



(a) Diversification with no domain knowledge

(b) Diversification with area knowledge

(c) Diversification with area and part knowledge

Fig. 11. Impact of domain knowledge on diversification (experiment 3)

6 Conclusion

The research described in this paper arises from the need to apply case-based diagnosis in a domain with very poor data quality. For this reason, conventional Case-Based Reasoning can only achieve a very low hit rate when the retrieval set size is not too large. Our solution is to use domain knowledge (which can be easily obtained in the current maintenance procedure) in the case matching process. We tested our solution on the real data obtained from the target company and performed an experimental study. Our results show that, the more domain knowledge is used, the higher the hit rate will be for both non-diversified and diversified approaches. Interestingly, diversification seems to become more effective when more domain knowledge is used.

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. 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
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. 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
8. 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
9. Zelikovitz, S., Hirsh, H.: Using LSI for Text Classification in the Presence of Background Text. In: Proceedings of the Tenth Conference for Information and Knowledge Management (2001) 113-118
10. 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
11. 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
12. 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
13. Zhang, L., Coenen, F., Leng, P.: Can Domain Knowledge Help Case Based Diagnosis? In: Proceedings of the Seventh UK CBR Workshop, 10 December (2002) 1-8