Personalized Web Document Classification using MCRDR

Sung Sik Park, Yang Sok Kim, and Byeong Ho Kang

School of Computing, University of Tasmania
Hobart, Tasmania, 7001, Australia

{sspark,yangsokk,bhkang}@utas.edu.au

Abstract. This paper focuses on real world Web document classification problem. Real world Web documents classification has different problems compare to experimental based classification. Web documents have been continually increased and their themes also have been continually changed. Furthermore, domain users' knowledge is not fixed apart from classification environments. They learn from classification experience, broaden their knowledge, and tend to reclassify pre-classified Web documents according to newly obtained knowledge to fit various contexts. To handle these kinds of problems, we use Multiple Classification Ripple-Down Rules (MCRDR) knowledge acquisition method. The MCRDR based document classification enables domain users to elicit their domain knowledge incrementally and revise their knowledge base (KB), and consequently reclassify pre-classified documents according to context changes. Our experiment results show MCRDR document classifier performs these tasks successfully in the real world.

1 Introduction

The size of available documents to be handled has grown rapidly since the Internet was introduced. For example, Pierre [1] estimates the number of pages available on the Web is around 1 billion with almost another 1.5 million added per day and some Internet search service companies reported that they cover around 3 billion pages [2]. Many Web document management systems have been developed because Web documents are now considered as one of the major knowledge resources.

Before computer technology was introduced, people mainly relied on manual classification such as library catalogue systems. In the early stages of the computerized classification development, computer engineers moved this catalogue system into the computer systems. However, as the size of available Web documents grows rapidly and people have to handle them within limited time, automated classification becomes more important.

Machine learning (ML) based classifiers have been widely used for automatic document classification and there are various approaches such as clustering, support vector

machine, probabilistic classifier, decision tree classifier, decision rule classifier, and so on[3]. But they have some problems when they are applied to real world applications because they capture only a certain aspect of the content and tend to learn in a way that items similar to the already seen items (training data) are recommended (predefined categories) [4]. However, it is difficult to collect well defined training data sets because Web documents (e.g., news articles, academic publications, and bulletin board messages) are continually created by distributed world-wide users and the number of document categories also continually increases. To manage this problem, the document classifiers should support incremental knowledge acquisition without training data. Though some ML techniques such as clustering techniques [5-7] are suggested as solutions for incremental classification, they do not sufficiently support personalized knowledge acquisition (KA). Document classifiers in the real world should support personalized classification because classification itself is a subjective activity [1]. To be successful personalized document classifiers, they should allow users to manage classification knowledge (e.g., create, modify, delete classification rules) based on their decision. But it is very difficult when users use ML classifiers because understanding their compiled knowledge is very difficult and their knowledge is so strongly coupled with the knowledge of training data sets that it is not easily changed without deliberate changing them.

Rule-based approach is a more favorable solution for the incremental and personalized classification task because the classification rules in knowledge base (KB) can be personalized, understood, and managed by users very easily. But rule-based systems are rarely used to construct an automatic text categorization classifiers since the '90s because of the knowledge acquisition (KA) bottleneck problem [3, 8]. We used Multiple Classification Ripple-Down Rules (MCRDR), an incremental KA methodology, because it suggests a way that overcomes the KA problem and enables us to use the benefits of rule-based approach. A more detail explanation will be suggested in section 2.

Our research focuses on the personalized Web document classifier that is implemented with the MCRDR method. In section 2, we will explain causes of the KA problem and how MCRDR can solve that problem. In section 3, we will explain how our system implemented in accordance with MCRDR method. In section 4, we will show empirical evaluation, which is performed three different ways. In section 5, we will conclude our research and suggest further works

2 Knowledge Acquisition Problems and MCRDR

KA problems are caused by cognitive, linguistic and knowledge representational barriers [8]. Therefore, the promising solution for the KA must suggest the methodology and KA tools that overcome these problems.

Cognitive Barrier. Because knowledge is unorganized and often hidden by compiled or ***tacit*** knowledge and it is highly interrelated and is retrieved based on the situation or some other external trigger, knowledge acquisition is discovery process. Therefore,

knowledge often requires correction and refinement - the further knowledge acquisition delves into compiled knowledge and areas of judgment, the more important the correction process becomes [9]. From the GARVAN-ES1 experience, Compton et al [10] provide an example of an individual rule that has increased four fold in size during maintenance and there are many examples of rules splitting into three or four different rules as the systems' knowledge was refined. Compton and Jensen[11] also proposed that knowledge is always given in context and so can only be relied on to be true in that context. MCRDR focuses on ensuring incremental addition of validated knowledge as mistakes are discovered in the multiple independent classification problems [12, 13].

Linguistic Barrier. Communication difficulties between knowledge engineers and domain experts are also one of the main deterrents of knowledge acquisition. Traditionally, knowledge is said to flow from the domain expert to the knowledge engineer to the computer and the performance of knowledge base depends on the effectiveness of the knowledge engineer as an intermediary [8]. During the maintenance phase, knowledge acquisition becomes more difficult not only because the knowledge base is becoming more complex, but because the experts and knowledge engineers are no longer closely familiar with the knowledge communicated during the prototype phase [11]. Domain knowledge usually differs from the experts and contexts. Shaw[14] illustrates that experts have different knowledge structures concerning the same domain and Compton and Jansen[11] show that even the knowledge provided by a single expert changes as the context in which this knowledge is required changes. For these reason, MCRDR shift the development emphasis to maintenance by blurring the distinction between initial development and maintenance and knowledge acquisition is performed by domain experts without helping the knowledge engineer[1] [13].

Knowledge Representation Barrier. The form in which knowledge is available from people is different from the form in which knowledge is represented in knowledge systems. The difference between them, called representation mismatch, is central to the problem of KA. In order to automate KA, one must provide a method for overcoming representation mismatch [15]. KA research has been aimed to replace the knowledge engineer with a program that assists in the direct "transfer of expertise" from experts to knowledge bases [16]. Mediating representation facilitate communication between domain expert and knowledge engineer. Intermediate representations provide an integrating structure for the various mediating representations and can form a bridge to the knowledge base[17]. We used ***folder structure user interface***, which is largely used for manual document classification in traditional document management application, as ***mediating representation method*** and ***difference lists*** and ***cornerstone cases*** as ***intermediating representation***. Folder manipulations are interrelated with the MCRDR KA activities in our system.

---

[1] This does not mean MCRDR needs no help from knowledge engineer or programmer. Rather, they are required for the initial data modeling (Kang, B. H., Compton, P., Preston, P., 1996).

3 Real World Web Document Classifier with MCRDR

The system, a text classification system for Web documents, is a component of the Personalized Web Information Management System (PWIMS) System [18]and is implemented with C++ program language and the MCRDR methodology. It is used to construct both Web document classification and personalized Web portal.

3.1 Folder Structures as a Mediating Representation

The choice of representation can have an enormous impact on human problem-solving performance [19, 20]. The term mediating representation is used to convey the sense of coming to understand through the representation and it should be optimized for human understanding rather than for machine efficiency. It is suggested to improve the KA process by developing and improving representational devices available to the expert and knowledge engineer. Therefore, it can provide a medium for experts to model their valuable knowledge in terms of an explicit external form [17]. We use traditional folder structures as a mediating representation because users can easily build a conceptual domain model for the document classification by using folder manipulation. Our approach differs from the traditional knowledge engineering approach because we assume there is no mediate person (knowledge engineer). Rather the domain experts or users directly accumulate their knowledge by using KA tools [12].

3.2 Inference with MCRDR document Classifier

A classification recommendation (conclusion) is provided by the last rule satisfied in a pathway. All children of satisfied parent rule are evaluated, allowing for multiple conclusions. The conclusion of the parent rule is only given if none of the children are satisfied [13, 21, 22]. For example, the current document has a set of keywords with {a, b, c, d, e, f, g}.

1. *The system evaluates all the rules in the first level of the tree for the given WL (rules 1, 2, 3 and 5 in Fig. 1.). Then, it evaluates the rules at the next level which are refinements of the rule satisfied at the top level and so on.*
2. *The process stops when there are no more children to evaluate or when none of these rules can be satisfied by the WL in hand. In this instance, there exist 4 rule paths and 3 classifications (classes 2, 5, and 6).*
3. *The system classifies into the storage folder structures (SFS)' relevant nodes (F_2, F_5, and F_6) according to the inference results.*
4. *When the expert finds the classification mistakes or wants to create the new classifications, he updates the classification knowledge via the knowledge acquisition interface.*

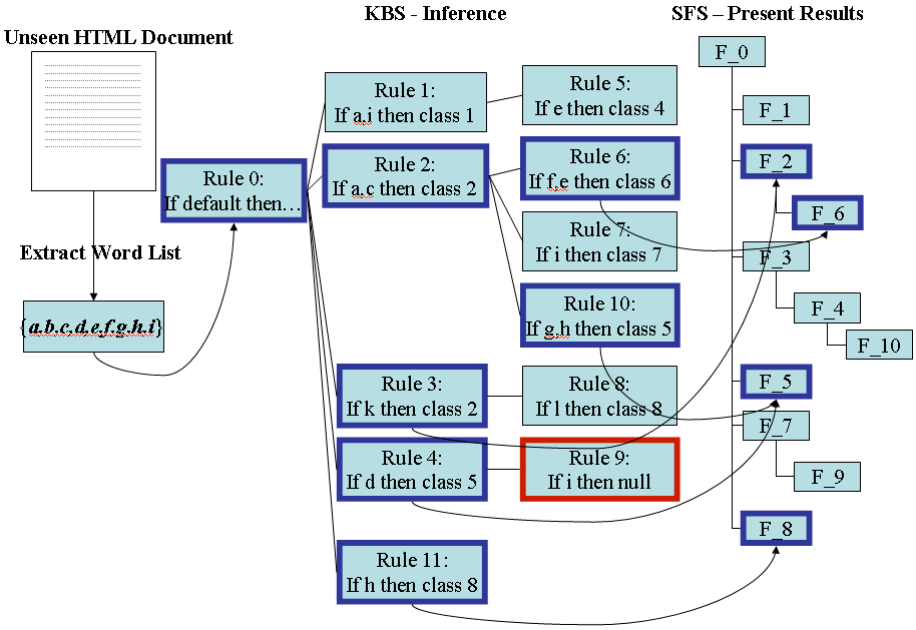

Fig. 1. Inference for the Web document classification

3.3 Knowledge Acquisition and Intermediate Representation

KA and inference are inextricably linked in the MCRDR method, so some KA steps depend on the inference and vice versa [13, 23, 24]. The KA process consists of the following sub-tasks: 1) initiating KA process, 2) deciding KA method, and 3) validating new rules.

Initiating KA Process. KA process is initialized by users when they dissatisfy the system's inference result. Kelly [25] suggested that "every construct has a specific range of convenience, which compromise all things to which the user would find its application useful." The range of convenience of each construct defines its extension in terms of a single aspect of a limited domain of events [17]. The users' decision for initializing new KA processes depends on the range of convenience. There are two different kinds of KA initialization: the KA process begins when the system recommends incorrect class or no class [23] and users initiate it (human initiated KA) and users move or copy some pre-classified documents to another folder (system initiated KA).

Deciding KA Methods. There are three kinds of KA methods: refinement KA, stopping KA, and ground-breaking KA.

- ***Refinement KA***: If the user thinks that the current document should be classified into the sub folder (may not exist) of the recommend folder, the user selects (or

creates and selects) the sub folder of the folder recommended by the system. The new rule should be added under the current classification rule as the child rule, because it refines current rule. For example, if a certain document that contains keyword "a" and "c", it will be classified into folder F_2 in Fig. 1. But users may want to classify this document to folder F_6 (this folder may not exist when this document classified) because it contains keyword "f" and "e". In this case, the new refinement rule is created under the rule 2 and its conclusion is class 6.

- ***Stopping KA***: If the current inference result is obviously incorrect and the users do not want to classify incoming documents into this folder, he/she makes stopping rules with certain condition keyword/keywords. The new stopping rule won't have any recommendation for a folder. For example, if a certain document that contain keyword "d", it will be classified folder F_5 in Fig. 1. But users may not want to classify this document to folder F_2 because it contains keyword "i". In this case, the new rule with condition "i" is added under the rule 4 and its conclusion is "null".
- ***Ground-breaking KA***: For example, if a certain document that contains keyword "k", it will be classified folder F_2 in Fig. 1. But domain experts may not want to classify this document to folder F_2 because it contains keyword "h" and they want to make new classification. In this case new rule is added under the root node (e.g. rule 11).

The KA process is initiated by system when users copy or move pre-classified documents to other folder/folders. Its KA method depends on the action types. If the action is moving, the stopping KA and ground breaking KA are needed. For example, if users want to move some documents in F_6 to F_1, they must select keywords that make stopping rules and ground breaking rules such as "***t***". In this example, new rule conditions will be "a" and "c" and "f" and "e" and "***t***". If the action is copying, only the ground breaking rule is automatically created by the system. Its condition is the same as the original rule but it has a different conclusion.

Validating with Cornerstone Case and Difference List. Bain [26] proposed that the primary attributes of intellect are consciousness of difference, consciousness of agreement, and retentiveness and every properly intellectual function involves one or more of these attributes and nothing else. Kelly[25] stated "A person's construction system is composed of a finite number of dichotomous construct." Gaines and Shaw[27] suggested KA tools that are based on the notion that human intelligence should be used for identifying differences rather than trying to create definitions. In our system, the experts must make domain decisions about the differences and similarities between objects to validate new rule. Our system supports users with ***cornerstone case*** and ***difference list*** [12, 13, 21, 23, 24]. As shown in Fig. 1, an n-ary tree is used for knowledge base (internal schema). MCRDR uses a "***rules-with-exceptions***" knowledge representation scheme because the context in the MCRDR is defined as the sequence of rules that were evaluated leading to a wrong conclusion or no conclusion with existing knowledge base [13]. Though users can see the whole knowledge base (internal schema) in our system, it is not directly used for KA. Instead, MCRDR uses ***difference list*** and ***cornerstone case*** for intermediate representation. The documents are used for the rule creation are called "**cornerstone cases**" and saved with the rules. Each folder may have multiple rules and cornerstone cases. When users make refinement rule or stopping rule, all related rules must be validated

but we do not want for users to make a rule that will be valid afterward. Rather we want to present the users with a list of conditions (called "***difference lists***") to choose from which will ensure a valid rule. The difference between the intersection of the cornerstone cases which can reach the rule and the new case cannot be used [12]. Cases which can be reclassified by the new rule appear in the system. The users may subsequently select more conditions from the different keywords lists to exclude these cases. Any case which is left in this list is supposed to classify the new folder by the new rule. A prior study shows that this guarantees low cost knowledge maintenance [13, 23].

4 Experiment

The goal of our research is to develop personalized Web document classifiers with MCRDR. The experiments are designed to the performance evaluation in the various classification situations. We consider three different cases: 1) document classification without domain change by single user, 2) document classification with domain changes by single user, and 3) document classification within single domain by two users.

Data Sets. We uses three different data sets: health information domain, IT information domain (English), and IT and finance domain (Korea), which are collected by our Web monitoring system for one month[18]. Table 1 represents the data sets that are used for our experiments.

Table 1. Inference for the Web document classification

Data	Domain	Source	User	Articles
Data Set 1	Health	BBC, CNN, Australian, IntelliHealth, ABC (US), WebMD, MedicalBreakthroughs	1	1,738
Data Set 2	IT (English)	Australian, ZDNet, CNN, CNet, BBC, TechWEB, New Zealand Herald	2	1,451
Data Set 3	IT/Finance (Korean)	JungAng, ChoSun, DongA, Financial News, HanKyeung, MaeKyeung, Digital Times, iNews24	1	1,246

Results. Classification effectiveness can be usually measured in terms of precision and recall. Generally two measures combined to measure the effectiveness. However, we only use precision measure because our system is a real world application and there is no pre-defined training data set. Fig. 2 shows the experiment's results. In each figure, horizontal axis represents the cases, left vertical axis represents the precision rates and right vertical axis represents the number of rules.

Experiment 1. This experiment is performed by a single user without domain changes in the health news domain. The user classified 1,738 articles with 348 rules. Though there are some fluctuations of the precision rate and rule numbers, there exist obvious trends: the precision rate gradually increases and the number of rules gradually decreases as the cases increase. Precision rate sharply increases from starting point

to a certain precision level (around 90%) and is very stable after that point. This is caused by the fact that the domain knowledge continually change and as the user knows the domain, the more classification knowledge is needed.

Experiment 2. This experiment is performed by a single user in IT and Finance news domain (Korean). Totally, 1,246 articles are classified and 316 rules are created by the users. At first, user classifies IT articles from the business relationship view (e.g. customers, competitors and solution providers). New view point for the domain (technical view) is added when user classifies 550 cases and new domain (finance) added when user classifies 800 cases. When the view point changed, the precision rate went down from 90% to 60% but precision rate recovers around 80% by classifying a small additional amount of cases. When the new domain (financial news) is added to the current domain (IT news), the precision rate sharply decreases to 10% and the rule creation goes up 30 but a very small number of cases is needed to recover 80% precision. This result shows that our document classifier can work efficiently with domain changes.

Experiment 3. This experiment is performed by two users in the same IT news domain (English). In total, they classified 1,451 articles with 311 rules: User 1 classifies 1,066 articles with 228 rules and user 2 classifies 432 articles with 83 rules. The classification result is shown in Fig. 2 (c). When user 1 classified 500 articles, the precision of classifier reached around 90%. After that point, new rules are gradually created and the precision rate is slightly improved until user 1 classifies 1000. When a different user (user 2) starts to classify, the precision rate shapely down to 60% and many new rules are created. But small articles are needed to get a similar precision rate. This result means that our classifier can be adaptively applied when different users classify.

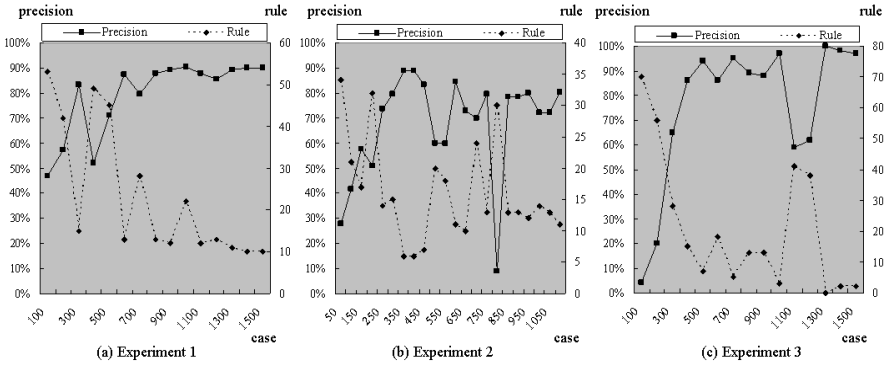

Fig. 2 Classification Results

5 Conclusion

We suggested the MCRDR based document classifier. MCRDR is an incremental KA method and is used to overcome the traditional KA problem. Our classifier used the traditional folder structures as a mediating representation. Users can construct their conceptual document classification structures by using an MCRDR based classifier. In

our system, the KA and inference process is inextricably linked, so some KA steps depend on the inference and vice versa. The KA process begins when the classifier suggest no folder or incorrect folders or users activate some function in folders such as copying or moving some cases. There are three different KA methods – refinement KA, stopping KA, and ground-breaking KA. In the validation process, we used corner cases and difference list as an intermediate representation. Experiment results show that users can create their document classifier very easily with small cases and our system successfully supports incremental and robust document classification. An incremental KA based classification works well in a certain domain where the information continually increases and the creation of training set for machine learning is hard.

However, this attitude does not deny the machine learning research works. Rather we view our approach can be a collaborator of machine learning technique. Wada et al. suggest integration inductive learning with RDR [28], Suryanto and Compton suggest a reduced KA with decision tree [29]. Especially we view our approach can help construct a fine training data set with cost efficiency in the initial stage. Research for the combining incremental KA approach with machine learning techniques will be our further work.

References

1. Pierre, J., *Practical Issues for Automated Categorization of Web Pages*. 2000.
2. Sullivan, D., *Search Engine Size*. 2003.
3. Sebastiani, F., *Machine Learning in Automated Text Categorization*. ACM Computing Surveys, 2002. **34**(1): p. 1-47.
4. Mladenic, D., *Text-learning and related intelligent agents: a survey*. IEEE Intelligent Systems, 1999. **vol.14, no.4**: p. 44-54.
5. Wong, W.-c. and A.W.-c. Fu. *Incremental Document Clustering for Web Page Classification*. in *IEEE 2000 Int. Conf. on Info. Society in the 21st century: emerging technologies and new challenges (IS2000)*. 2000. Japan.
6. Liu, R.-L. and Y.-L. Lu. *Incremental context mining for adaptive document classification*. in *Conference on Knowledge Discovery in Data*. 2002: ACM Press New York, NY, USA.
7. Charikar, M., et al. *Incremental clustering and dynamic information retrieval*. in *Annual ACM Symposium on Theory of Computing*. 1997. El Paso, Texas, United States: ACM Press New York, NY, USA.
8. Musen, M.A., *Automated Generation of Model-Based Knowledge-Acquisition Tools*. Research Notes in Artificail Intelligence. 1989, San Mateo, CA: Morgan Kaufmann Publishers, Inc.
9. Lawrence K, L., *Collision-Theory vs. Reality in Expert Systems*. 2nd ed. 1989, Wellwsley, MA: QED Information Sciences, Inc.
10. Compton, P., et al. *Maintaining an Expert System*. in *4th Australian Conference on Applications of Expert Systems*. 1988.
11. Compton, P. and R. Jansen, *A philosophical basis for knowledge acquisition*. Knowledge Acquisition, 1990. **vol.2, no.3**: p. 241-258.

12. Compton, P., et al., *Knowledge acquisition without analysis*. Knowledge Acquisition for Knowledge-Based Systems. 7th European Workshop, EKAW '93 Proceedings, 1993: p. 277-299.
13. Kang, B.H., P. Compton, and P. Preston, *Validating incremental knowledge acquisition for multiple classifications*. Critical Technology: Proceedings of the Third World Congress on Expert Systems, 1996: p. 856-868.
14. Shaw, M.L.G. *Validation in a knowledge acquisition system with multiple experts*. in *the International Conference on Fifth Generation Computer Systems*. 1988.
15. Gruber, T.R., *Automated knowledge acquisition for strategic knowledge*. Machine Learning, 1989. **vol.4, no.3-4**: p. 293-336.
16. Davis, R., *Applications of Meta Level Knowledge to the Construction, Maintenance, and Use of Large Knowledge bases*. 1976, Stanford University: Stanford, CA.
17. Ford, K.M., et al., *Knowledge acquisition as a constructive modeling activity*. International Journal of Intelligent Systems, 1993. **vol.8, no.1**: p. 9-32.
18. Park, S.S., S.K. Kim, and B.H. Kang. *Web Information Management System: Personalization and Generalization*. in *the IADIS International Confernece WWW/Internet 2003*. 2003.
19. Hahn, J. and J. Kim. *Why are some representations (sometimes) more effective?* in *20th International Conference on Information Systems*. 1999. Charlotte, North Carolina, United States: Association for Information Systems Atlanta, GA, USA.
20. Larkin, J. and H. Simon, *Why a diagram is (sometimes) worth ten thousand words*. Cognitive Science, 1987. **11**: p. 65-99.
21. Compton, P. and R. D. *Extending Ripple-Down Rules*. in *12th International Conference on Knowledge Engineering and Knowledge Managements (EKAW'2000)*. 2000. Juan-les-Pins, France.
22. Martinez-Bejar, R., et al., *An easy-maintenance, reusable approach for building knowledge-based systems: application to landscape assessment*. Expert Systems with Applications, 2001. **vol.20, no.2**: p. 153-162.
23. Kang, B.H., W. Gambetta, and P. Compton, *Verification and validation with ripple-down rules*. International Journal of Human-Computer Studies, 1996. **vol.44, no.2**: p. 257-269.
24. Compton, P. and D. Richards, *Generalising ripple-down rules*. Knowledge Engineering and Knowledge Management Methods, Models, and Tools. 12th International Conference, EKAW 2000. Proceedings (Lecture Notes in Artificial Intelligence Vol.1937), 2000: p. 380-386.
25. Kelly, G.A., *The Psychology of Personal Constructs*. Vol. 1. 1955, NY: W. W. Norton & Company Inc.
26. Bain, A., *Mental and Moral Science*. 3rd ed. 1884, London: Longmans, Green, And Co.
27. Gaines, B.R. and M.L.G. Shaw. *Cognitive and Logical Foundations of Knowledge Acquisition*. in *5th AAAI Knowledge Acquisition for Knowledge Based Systems Workshop*. 1990.
28. Wada, T., et al. *Integrating Inductive learning and Knowledge Acquisition in the Ripple Down Rules Method*. in *6th Pacific Knowledge Acquisition Workshop*. 2000. Sydney, Australia.

29. Suryanto, H. and P. Compton. *Intermediate Concept Discovery in Ripple Down Rule Knowledge Bases*. in *2002 Pacific Rim Knowledge Acquisition Workshop*. 2002. Tokyo, Japan.