A Data Quality Index with Respect to Case Bases within Case-Based Reasoning

Jürgen Hönigl and Josef Küng

Institute for Application-Oriented Knowledge Processing
Johannes Kepler University
Linz, Austria
{juergen.hoenigl,josef.kueng}@jku.at

Abstract. Within Case-Based Reasoning (CBR), terms concerning quality of a case base are mentioned in publications, but partially without clarifications of criteria. When developing a CBR system from scratch, an index for case base quality supports an assessment of the actual cases. In this approach, both theory and an application are demonstrated. An index was defined and subsequent applied within a current CBR project, which is under development. In addition, various approaches concerning case base quality are demonstrated. Big data occurs within a combination of high velocity, great volume and variety of incoming data. Defining an index to measure the case base quality copes with that.

1 Introduction

Within this section, the introduction was divided into several parts to demonstrate the motivation, a few statements about CBR and an outline.

1.1 Motivation

When reading literature about case-based reasoning, it was written about the quality of a case base and avoiding too redundant cases within case base. Various approaches are existing but partially with fuzzy definitions and primarily without clear results. Especially when researching towards an eventual re-use of an index. Therefore, the authors were defining an index to describe case base quality. This was applied within the first author's doctoral thesis as part within the proof of concept. Closing the gap between big data and CBR can be seen as a drive towards an easy to apply index for new relevant cases with respect to the size of a case base. The significance of a data quality index can be seen within the next annotations.

1.2 Significance towards a Case Base

A case base contains knowledge, which will be used for the reasoning process of a case-based reasoning system. An index, which states the quality of a case

N.T. Nguyen et al. (Eds.): ACIIDS 2014, Part I, LNAI 8397, pp. 432–442, 2014.

base, can be used within different steps of the CBR model given by Aamodt and Plaza.[1] A deletion strategy for too similar cases has to applied to a CBR system to keep the quality of a case base. A deletion strategy is one possible point to deal with the size of case base concerning the maintenance. Another point of view, establish rules for pre-processing to avoid not suitable reasoning efforts and impaired cases. For instance, a typo could cause an impaired case when not using pre-processing assertion rules. A customer with an age of 92 years (instead of 29 years) could be reasoned within a CBR system, but it would be an outlier within the case base. Subsequent, this case would be removed according to a deletion strategy, which uses the not recently used paradigm for instance. Within CBR, applying an index can combined with committing a database state. When receiving many new cases within a CBR approach, the advantage of an index can be seen to init a rollback of the database state, which reflects the case base, according to a modified index value with a percentage of minus 20 for instance. Big data occurs if a great volume, a high velocity and variety (structured and unstructured data) will be received. Even two of them can decrease the quality of a case base. A great volume of data with a high velocity can contain too many redundant and obsolete cases. Within a CBR system, pre-processing and similarity measures can avoid many inadequate data, but an assessment of the case base has to be applied in addition. When working on case mining, a complete case base without missing values should be seen as a pre-condition. For instance, gaining association models requires complete cases.[10] When considering an evolution such as IBM's (Industrial Business Machines) research projects Watson and DeepBlue within a decade, it is obvious that these projects can cope with missing values within their knowledge bases.[6], [14] In contrast, a CBR approach requires data within the case base because a CBR system is not intended to implement various application programming interfaces to download information on the fly.[12] In addition, the knowledge base of IBM's Watson contained a huge amount of text volumes, databases and journals.[7], [9]

1.3 Outline

To briefly present a red line regarding this paper, firstly, related work is demonstrated. Then, three sub-indices are demonstrated, which are required, to build the main index of this approach. Subsequent, the index will be calculated on a top level. Afterwards the application of the index will be explained within a case-based reasoning prototype. Subsequent, a discussion is presented regarding various sights when using thresholds for instance. At the end, a conclusion and eventual future work are enumerated.

2 Related Work

This section demonstrates chronological various possibilities concerning the term case base quality within literature. In 1997, an approach was stated to combine

decision theory and CBR. This idea could be used if many missing values would occur to use CBR together with decision theory within an area like unfinished alternatives. Therefore, considering of quality weakness within a case base could be compensated. On the other side, their approach was an experiment and explained difficulties when combing two kind of decision support technologies. For instance, they have detected obstacles when using normative models due to the application of probability and utility for preference and judgement in combination with CBR.[19]

A historic approach given in 1998 refers to non-functional requirements regarding CBR systems. Their approach was applied within the medical domain. The efforts made were primarily focused on a CBR system instead of the managed data. An intersection between their system-related approach and a data-related approach can be seen within their work on confidentiality and integrity of data.[11]

The quality improvement paradigm (QIP) refers to steps to consider when developing a CBR system. Basili presents a cycle to gain a good combination of technical and managerial solution to achieve a professional CBR application development. The experience factory refers within various steps to different issues, which seems like a waterfall structure at first sight. However, these steps can be partially used in an iterative way, which avoids that. To give a brief explanation concerning this paradigm, two quality-related steps are stated. Within *characterize* (QIP1), the scope of the project will be defined, which results into a context for a goal definition. In addition, experience from the experience base can be selected. The experience base is a knowledge base of past projects related to achieved experience. *Set goals* (QIP2) consider different viewpoints such as customer, project manager and user. The defined goals must be measurable.[4]

Within an old approach presented in 2000, quality measures were defined to assess the case base quality with criteria such as correctness, consistency, uniqueness, minimality, and incoherence. They implemented their approach within a framework, but there is a lack concerning eventual other projects when considering application of their approach. In addition, they clearly stated that similarity measures would improve the performance of their assessment. On the other hand, clustering was defined as an issue to perform if their assessment would not be able to process too many cases *in a reasonable amount of time.*[17]

Within an approach concerning the maintenance, existing CBR approaches were applied to summarize them into a new approach. On the basis of the Aamodt and Plaza approach [1] and various INRECA research activities [5], terms were reused and combined. They divided their theoretic generic approach into three stages named retain, review and restore. For instance, retain refers to complete a case. Review points to an assessment of a case and restore implies modifying a case.[18]

Within INRECA (Induction and Reasoning from Cases), case base quality was mentioned, but not concrete stated within a definition of eventual solutions. For instance, a term like *define clear objectives* sounds too unclear to consider

it within a concrete index towards case base quality from the authors point of view.[5]

Another approach tried to solve and improve maintenance issues with CBR classifiers. They used clustering and logistic regression to build their classifiers. Their approach was not applied within a generic way. Apart of that, the adaptation feature was neglected. Assigning a string label was their *simple adaptation*. When having the focus on maintenance, then adaptation must be carefully integrated into a CBR system from the authors point of view.[2]

An approach namely *Assessing Case Base Quality* states interesting notes, but some critical points towards their approach could be seen such as a missing portability and too much effort to integrate their approach. Their main goals were to assess and measure inherent problem-solution irregularity within a case base to improve using cases especially with respect to the accuracy concerning solutions. The Mantel Test (or Mantel's Randomisation Test) was applied together with different ratios to assess the quality of their case base. Therefore, their approach was not implemented in a generic way.[16]

Within [15], they stated an approach towards a case-mining algorithm. This generates a *competent* case base when processing existing cases. The stated two issues within their approach. On the one hand, processing nearest cases, which are not containing correct solutions. And another point of view, an uneven case distribution was named as potential obstacle. In addition, they proposed an algorithm to mine within cases, which includes avoiding the previous mentioned problems. Concerning their case-mining approach, they stated two points, which are worth to mention. With respect to the approach in this paper and their approach, their points are overlapping concerning an idea behind when searching for an intellectual intersection between different approaches.

- *Each case should cover as much of the problem space as possible to reduce the potential bias, and*
- *The cases should be as diverse as possible to reduce co-variance in producing errors.*

[15] When reading these items, a brief comparison to the quality index can be made. The first item above can be seen as avoiding missing values within this approach (third sub-index) concerning an index. The second item above can be seen within similar retained queries in this approach (second sub-index). In addition, the second item above can be partially seen within the first sub-index when assessing average solutions per case.

3 Building Sub-indices

Three indices are used to build an index for the quality of case base. Each of these sub-indices uses an interval from 0 to 1.

3.1 Index I: Average Solutions per Case

When using a revision graph for solutions, then an entire revision graph will be defined as 1 solution concerning this index. Null adaptation implies only one

solution for a problem, but using a revision graph implies more than one solution for a query. At the end, only one solution is defined as an actual solution for a problem when using a revision graph. Therefore, using revision graphs must not aggravate this index. Multiple solutions are considered as an additional processing effort. In addition, maintenance of a case base can be more difficult with increasing similar solutions. A threshold concerning the maximum number of solutions per case has to be defined within a theoretical interval [1,count of solutions]. A practical interval would be from 3 to 9. For each case, the count of bad solved cases (argument *cc*), concerning too many solutions, will be incremented if the given threshold was reached. Subsequent, the sub-indices can be calculated with respect to all cases (argument *c*).

$$Idx_I = 1 - \frac{cc}{\sum c} \tag{1}$$

3.2 Index II: Count of Similar Retained Queries

To define similar retained queries, a similarity measure has to be applied with a certain threshold. A problem to problem similarity measure must exist with a known interval to define a threshold for a case base. If a threshold was reached, then the count has to be incremented. Subsequent, an index can be calculated with following formulae:

$$Idx_{II} = 1 - \frac{csrq}{\sum qc} \tag{2}$$

The count of similar retained queries is given by argument *csrq* and the query comparisons are denoted as *qc*.

3.3 Index III: Missing Values

The count of missing values (*cmv*) within cases, with respect to the count of occurrence, has to be calculated. The actual sum of fields (f) can be achieved within the persistence of a case base when counting all table fields.

$$Idx_{III} = 1 - \frac{cmv}{\sum f} \tag{3}$$

4 Calculating the Main Index

To clearly state the formulae, this section presents the integration of the three sub-indices stated above.
The case base quality index (CBQ) uses an interval from 0 to 100. 100 per cent states the best possible value for a case base and 0 per cent refers to a impaired value of a case base. The previous mentioned indices are subsequently weighted.

$$CBQ = 100 \cdot \frac{Idx_I \cdot Weight_I + Idx_{II} \cdot Weight_{II} + Idx_{III} \cdot Weight_{III}}{\sum_{i=1}^{i=3} Weight_i} \tag{4}$$

The weight factors can be applied concerning a concrete case base within a given domain. For instance, if avoiding of missing values is more important than the case redundancies, then $weight_{III}$ will receive another argument in comparison to $\frac{1}{3}$.

5 Application of the Index within Loaner

This section covers the practical aspects of the implementation regarding the index described above. Within code name Loaner, an application written in C# and LINQ (Language Integrated Querying), the approach of this paper was implemented. The visualization was made when using Windows Presentation Foundation (WPF). The training set of the data was analyzed due to the actual implementation state.[8] It is complete and without multiple solutions, which refers to a good value concerning the case base quality.

5.1 I - Solutions per Case

The used threshold for solutions per case was 7. Zero cases are reached this threshold. This generates a value of 1.

5.2 II - Similar Retained Queries

The chosen threshold was defined as 80 per cent. This was detected within prior experiments based on development of similarity measures. When using a high value such as 95 per cent, zero similar queries would occur. Within the screenshot of Loaner, a page depicts the counting process of sub-index II. 28 similar retained queries were achieved within 498501 query comparison iterations. This implies a temporal value of $\frac{28}{498501}$, which will be subsequently subtracted from 1. Therefore, the value within this sub-index results into $\frac{498473}{498501}$.

Fig. 1. Loaner 0.4 α - Measurement Index II

5.3 III - Missing Values

In fact, the train set of the actual approach is complete concerning the values. Each tuple contains a value for each column. Zero missing values occurred within the data. This generates an excellent sub-index III, value 1.

5.4 Using the Main Index

To avoid to fall into oblivion, the train set is complete without identical cases. This refers to a high quality concerning the case base in prior to an assessment of the quality.

$$CBQ = 100 \cdot \left(1 \cdot \frac{1}{3} + \frac{498473}{498501} \cdot \frac{1}{3} + 1 \cdot \frac{1}{3}\right) \tag{5}$$

In this application, the case base quality index refers to 99.9981277202.

5.5 Experiments with Weights

Weights were considered for similarity measures and the formulae above.

In experiments concerning similarity measures, it was observed that only the attribute gender should be weighted with $\frac{1}{3}$. Otherwise, a simple similarity measure, which uses only a few attributes could increase or decrease the value of the result too much. Therefore, all attributes (except gender) are using the weight 1.

All sub-indices were associated with a weight of $\frac{1}{3}$ within the main index. In this case, increasing the weight for sub-index II would decrease the index value. Another point of view when consider additional data with missing values, this would wrongly increase the index value. Therefore, a cautious weighting was applied. When using another weight for sub-index II such as $\frac{5}{6}$, the value of main index is marginally modified to 99.9953193006. $\frac{5}{6}$ would be a too high value for a sub-index, but in this case the result of the main index is not really affected because the associated value of the sub-index was rather high $\left(1 - \frac{28}{498501}\right)$.

6 Discussion

This section provides a few notes about circumstances concerning the prototype Loaner and explanations with respect to the quality index. Concerning sub-index III, the natural assumption for this index is that an application code prevents to store cases with primarily null values. Otherwise bad case-based reasoning results would occur beside of low values in sub-index III. Within an interval [0,100], thresholds were tested against the case base to see various similarity values. Within the diagram, thresholds and an associated count of similar query comparisons are presented. The ordinate presents the count of query comparisons from 0 to 498501. The abscissa presents thresholds from 0 to 100.

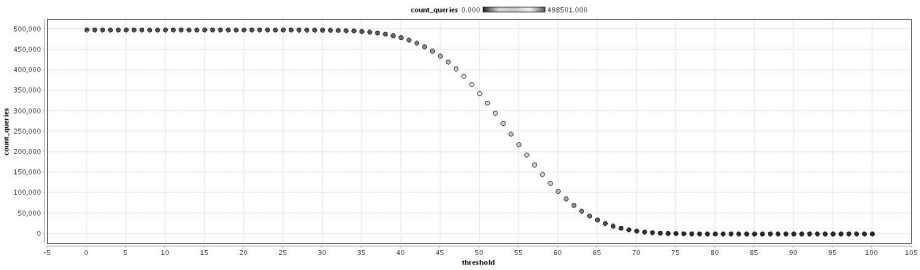

Fig. 2. Plot Thresholds 0 to 100

Within the threshold interval [0,100], the plot above presents that 57 per cent are a point to distinguish between the nearest queries and not related queries. Concerning sub-index II, 80 per cent was used because a threshold lower than 60 per cent would deliver many queries related to the concrete example within Loaner. For instance, the threshold 57 per cent refers to a count of 168570 queries. To use an adequate threshold for sub-index II, the concrete data such as a comma separated value file has to be analyzed. To give an excerpt within the higher threshold values regarding the second sub-index, a few relations are stated as follows.

- Threshold $\rightsquigarrow$ Count queries
- 75 $\rightsquigarrow$ 710
- 76 $\rightsquigarrow$ 407
- 77 $\rightsquigarrow$ 220
- 78 $\rightsquigarrow$ 117
- 79 $\rightsquigarrow$ 65
- 80 $\rightsquigarrow$ 28
- 81 $\rightsquigarrow$ 11
- 82 $\rightsquigarrow$ 6
- 83 $\rightsquigarrow$ 3
- 84 $\rightsquigarrow$ 2
- 85 $\rightsquigarrow$ 0

In addition, it is clearly presented that a percentage of 100 refers to zero similar retained queries. Therefore, 100 per cent is not suitable as threshold when using a similarity measure. Another point of view, a similarity with 100 per cent would be identical tuples, which has to be avoided when inserting data into a schema. In the second scatter plot, thresholds within the range [50,85] are depicted, which states an excerpt of the first scatter plot. The count of similar query comparisons starts with 0 and ends with 343038. When comparing this range to the full query range within the first scatter plot, it is clearly stated that within the range [50,85] a higher variability occurs concerning the similar query comparisons.

The second scatter plot presents that similarity values are reduced with various different steps in a range 50 to 85. Within Loaner, different similarity measures are using various attributes. For gaining the similarity value concerning

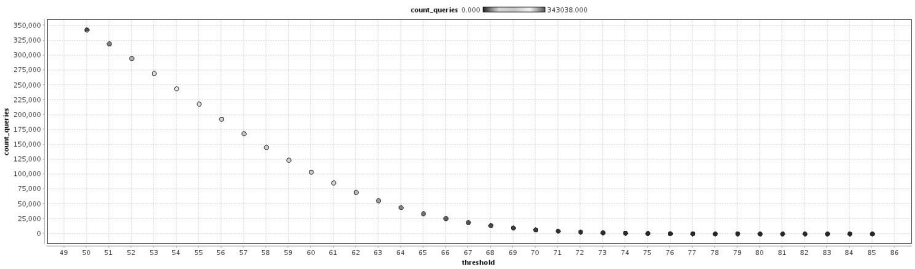

Fig. 3. Scatter Plot Thresholds 50 to 85

sub-index II, a similarity measure was applied, which uses all attributes. Those are age, credit amount, credit duration, number of people liable, other installment plans, gender, personal state, purpose of the loan, credit history, employment duration, job level, other credits, duration of the current residence, installment rate concerning disposable monthly income to give an excerpt. When using all attributes, no aspect such as personal-related issues (age, gender) or credit-related considerations (credit history, credit amount) will be neglected. Sub-index II calculated 28 similar retained queries within 498501 unique comparisons between different queries. Identical tuples are not persisted. Reflexive comparisons are avoided. Double comparisons are avoided in addition. For instance, the similarity between query id 100 and id 770 is calculated, but not vice versa.

A second data set was integrated into Loaner and analyzed. The additional data set was retrieved by the author of [3]. It was more numeric-based and contained more tuples in comparison to the first one. At first sight, the data set was evaluated with the same similarity measures which are applied towards the German data set. It contained no redundant solutions per case, only marginal similar retained queries and no missing values.

$$CBQ = 100 \cdot \left(1 \cdot \frac{1}{3} + \left(1 - \frac{193931}{4871881}\right) \cdot \frac{1}{3} + 1 \cdot \frac{1}{3}\right) \tag{6}$$

Hence, the result was stated as 98.6731271419 per cent. For determining the similar retained queries, 80 per cent was applied again as similarity value towards all query comparisons within the case base — except reflexive and redundant (id 31⇝94 but without 94⇝31 for instance) comparison steps.

7 Conclusion

Within Loaner, the application regarding sub-indices I and III was fatly achieved due to a complete training set. Sub-index II required an implementation, which refers to similarity measures. To avoid overlooking about similarities within queries, all attributes are applied to consider different aspects within a loan

application. Concerning the theory, the three sub-indices are easy to use. When using weighting with the index formulae described above, agility can be attached to fit specific requirements of a given domain. In this approach, the weighting of the sub-indices within the formulae above was stated with $\frac{1}{3}$. For sub-index II, a generic threshold cannot be inferred due to many different domains, which are suitable for case-based reasoning. These are car mechanic, structural health monitoring, employee support, call center tools and text retrieval software for instance to refer to this diversity. To infer this approach within three steps namely The Good, the Bad and the Ugly.[13]

- The Good - it clearly presents an index within a defined interval [0,100]
- the Bad - even a generic index needs implementation effort
- the Ugly - using wrong weights to hide weakness of a case base would be possible

Big data can be applied to CBR, but not using an index concerning the case base quality could lead to obstacles. Especially if a deletion strategy was not applied within a CBR approach. A case base with redundant and unused cases impairs the performance in reasoning processes. The proposed index can be applied to prevent these performance obstacles.

8 Future Work

When a loan application simulator will be finished, the case base quality index can be applied to new cases for further testing with weights. Apart of that: An automatic evaluation feature can be implemented to avoid outlier values for an index. For instance, detection of bad used weighting when using a weight such as $\frac{7}{10}$ for excellent managed similar retained cases when applying a weight like $\frac{1}{10}$ for too many missing values.

Acknowledgement. Appreciation goes to the reviewers for their comprehensive hints and feedback. Thanks to Prof. Baesens (who was not a reviewer) for providing an additional data set to enhance the research for both thesis and this paper.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. 7(1), 39–59 (1994)
2. Arshadi, N., Jurisica, I.: Maintaining case-based reasoning systems: A machine learning approach. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 17–31. Springer, Heidelberg (2004)
3. Baesens, B., Setiono, R., Mues, C., Vanthienen, J.: Using neural network rule extraction and decision tables for credit-risk evaluation. Management Science 49(3) (2003)
4. Basili, V.R.: The experience factory: Packaging software experience (1999)

5. Bergmann, R., Althoff, K.-D., Breen, S., Göker, M.H., Manago, M., Traphöner, R., Wess, S.: Developing Industrial Case-Based Reasoning Applications, 2nd edn. LNCS (LNAI), vol. 1612. Springer, Heidelberg (2003)
6. DeCoste, D.: The future of chess-playing technologies and the significance of kasparov versus deep blue. Papers from the 1997 AAAI Workshop (1997)
7. Ferrucci, D.A.: Ibm's watson/deepqa. SIGARCH Computer Architecture News 39(3) (2011)
8. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
9. Hönigl, J., Kosorus, H., Küng, J.: On reasoning within different domains in the past, present and future. In: 23rd Database and Expert Systems Applications (DEXA), 2nd International Workshop on Information Systems for Situation Awareness and Situation Management - ISSASiM 2012 (September 2012)
10. Hönigl, J., Nebylovych, Y.: Building a financial case-based reasoning prototype from scratch with respect to credit lending and association models driven by knowledge discovery. In: Central & Eastern European Software Engineering Conference in Russia (November 2012)
11. Jurisica, I., Nixon, B.A.: Building quality into case-based reasoning systems. In: Pernici, B., Thanos, C. (eds.) CAiSE 1998. LNCS, vol. 1413, pp. 363–380. Springer, Heidelberg (1998)
12. Leake, D.B.: Cbr in context: The present and future. In: Reasoning From Reminding, pp. 3–30. MIT Press (1996)
13. Leone, S.: The good, the bad and the ugly. il buono, il brutto, il cattivo (original title) (1966)
14. Newborn, M., Newborn, M.: Deep blue establishes historic landmark. In: Beyond Deep Blue, pp. 1–26. Springer, London (2011)
15. Pan, R., Yang, Q., Pan, S.J.: Mining competent case bases for case-based reasoning. Artificial Intelligence 171(16-17), 1039–1068 (2007)
16. Rahul Premraj, M.S.: Assessing case base quality. Bournemouth University and Brunel University (2005)
17. Reinartz, T., Iglezakis, I., Roth-Berghofer, T.: On quality measures for case base maintenance. In: Blanzieri, E., Portinale, L. (eds.) EWCBR 2000. LNCS (LNAI), vol. 1898, pp. 247–260. Springer, Heidelberg (2000)
18. Roth-Berghofer, T., Reinartz, T.: Mama: A maintenance manual for case-based reasoning systems. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 452–466. Springer, Heidelberg (2001)
19. Tsatsoulis, C., Cheng, Q., Wei, H.Y.: Integrating case-based reasoning and decision theory. IEEE Expert 12(4), 46–55 (1997)