

What Evaluation Criteria Are Right for CCBR? Considering Rank Quality

Steven Bogaerts and David Leake

Computer Science Department, Indiana University, Lindley Hall 215
150 S. Woodlawn Avenue, Bloomington, IN 47405, U.S.A.
{sbogaert, leake}@cs.indiana.edu

Abstract. Evaluation criteria for conversational CBR (CCBR) systems are important to guide development and tuning of new methods, and to enable practitioners to make informed decisions about which methods to use. Traditional criteria for evaluating CCBR performance by *precision* and *efficiency* provide useful information, but are limited by their focus on the single point at which a case is selected at the end of the system dialogue, and by their dependence on a model of the user's case selection criteria. This paper begins by revisiting issues in the evaluation of CCBR systems, arguing for the value of assessing the quality of the intermediate dialogue before case selection. It then proposes an evaluation approach based on *rank quality* to provide a fuller picture of system performance, and illustrates with an empirical study the use of rank quality to illuminate characteristics of similarity assessment strategies for partially-specified cases.

1 Introduction

Conversational case-based reasoning (CCBR) is an interactive paradigm in which situation assessment is done incrementally in a dialogue with the user. Because CCBR is extensively used in CBR applications (e.g., [1]), having the right criteria for evaluating the CCBR process is crucial, both scientifically and practically, for guiding developers and practitioners in system tuning. At each step of the basic CCBR cycle, the system presents the user with a set of potentially relevant cases and questions; the user may either select a question to answer or terminate the dialogue by selecting a case. Each time the user answers a question, the answer adds to the system's problem description, and the system generates a new candidate list. Because CCBR can be seen as aiming to rapidly drill down to a relevant case, influential work by Aha and Breslow [2] proposed evaluating CCBR systems based on *precision*, which measures whether the solution of the selected case adequately resolves the target problem, and *efficiency*, which measures the number of questions that are asked before a candidate case is selected.

Precision and efficiency criteria focus on a snapshot at the time of case selection, and do not reflect properties of the intermediate dialogue such as how consistently the system's suggested cases converge towards the final ranking. Such information may be especially important to assess as CCBR expands beyond traditional diagnostic tasks into new areas such as product recommendation, in which the initial dialogue may affect final user preferences. In addition, assessing precision and efficiency requires having a

model of which cases the user will select, which—as results in this paper demonstrate—may strongly influence evaluation results. To address these issues, this paper proposes an alternative approach to CCBR system evaluation, *rank quality*, which assesses how well the list of system-proposed cases at each step approximates the list of cases that would be generated if a complete problem description were available. This is meaningful at any point in the dialogue, and can be evaluated independent of the user’s case selection criteria.

Despite the intuitive appeal of the rank quality approach, formalizing rank quality involves surprisingly subtle issues. This paper briefly illustrates some of these issues, presents a rank quality criterion designed to address them, and defines a property of case bases, *distance granularity*, which can help determine the suitability of the defined rank quality metric for a particular case base.

The paper then presents an experimental examination of characteristics of precision and efficiency compared to rank quality in practice, for CCBR systems using five different similarity assessment strategies for partially-specified cases from Bogaerts and Leake [3], applied to three datasets from the UCI archive [4]. The experiments demonstrate the sensitivity of precision–efficiency approaches to the case selection model and illustrate how the rank quality approach can provide useful information about characteristics of the overall dialogue, illuminating differences in candidate similarity assessment strategies. This makes rank quality a promising tool for guiding the choice of similarity assessment strategies during system development. The paper closes by placing the results in context of other approaches to evaluating CCBR systems.

2 Precision, Efficiency, and Rank Quality Measures

Precision and efficiency are useful because they address two central concerns for CCBR: to identify a case which solves the current problem (as measured by precision), and to do so rapidly (as measured by efficiency). These measures are normally calculated by simulation experiments, based on a model of user behavior. The new approach proposed here, rank quality, quantifies the degree to which the list of candidate cases provided by the CCBR system at the current point in the dialogue matches the list that would be retrieved if all information about the current problem were known.

More precisely, let t be a full description of a target problem (i.e., a description in which all attribute values that will be revealed in the dialogue are already known). Let \hat{t} represent the current incomplete state of that problem description, under development in a CCBR dialogue. Let L be the set of possible ordered lists of cases presented by the CCBR system to the user, and let L_d be the ordered list of cases presented by the CCBR system to the user when the currently-known problem attributes correspond to description d .¹ The rank quality value is $c(L_t, L_{\hat{t}})$, for L_t the ideal list, $L_{\hat{t}}$ the current candidate list, and $c : L \times L \rightarrow [0, 1]$ a list order comparison function. We note that this formulation of the rank quality calculation depends on having access to a fully known target problem; it is intended to be applied in experimental settings in which such

¹ Here we assume that retrieval will depend only on the attributes in d , not on the order in which those attributes were revealed to the system. Adjusting the definition to allow for order-dependent case selection would not affect the substance of the definition of rank quality.

information is available, e.g., when testing alternative similarity assessment strategies during system development. A topic for future research is how rank quality might be applied to measure system performance up to the current point in an actual dialogue, e.g., by assuming the current problem description is "fully known".

To capture intuitions concerning rank quality, the value of c should increase monotonically with the "similarity" of the ordering of cases in the lists, with $c(l, l) = 1$ for any $l \in L$. However, to actually define an appropriate function is surprisingly subtle. Section 3 discusses general motivations for rank quality, while assuming that a function with intuitive behavior is available, and Section 4 proposes a formal definition.

3 Motivations for a Rank Quality Approach

Compared to precision and efficiency, rank quality approaches bring two primary benefits: (1) Providing a fuller picture of system behavior, because they can be applied at any point in a CCBR dialogue and because they assess the entire candidate list, and (2) not requiring assumptions about the user's criteria for final case selection.

Removing the Need for Selection Criteria Assumptions: Because precision and efficiency can only be determined at case selection time, automated evaluations of these properties typically gather performance statistics for a simulated user. These statistics are often gathered in either *leave-one-out* trials, in which a given case from the case base provides the target problem and the correct solution, and is removed from the case base for the duration of the trial, or *leave-one-in*, in which the target case remains in the case base [2]. At each step in the dialogue, the simulated user either selects a question to answer according to the target problem, or terminates the dialogue by selecting a suggested case. Case selection may be triggered, for example, when the similarity of a candidate case exceeds a threshold, or when no unasked questions remain. As we show in Section 5, precision and efficiency results can depend strongly on the specific user (case selection) model chosen. This dependence is problematic for assessing CCBR systems, because there are no obvious criteria for settings to use in such tests. McSherry [5] has shown that it is sometimes possible for a system itself to automatically terminate the dialogue without loss of solution quality, but a user might still choose to terminate the dialogue early, or might choose a suboptimal case. Consequently, the user model plays an important role in evaluation. To our knowledge, no human-subjects studies have systematically evaluated the case selection process for different subject populations. Even if such studies were done, a developer might lack information on the likely user population for a specific system. Because rank quality is based on a comparison of alternative system outputs, independent of the user, it removes the need for a case selection model.

Ability to Assess the Dialogue Instead of the Single Selected Case: Another benefit of the rank quality approach is the ability to provide information about how the system performed at each point during the CCBR dialogue. We expect the ability of rank quality to assess the quality of a set of intermediate suggestions to be useful to system designers because of how case ordering in intermediate steps may affect user confidence, the

user's ability to make the right decisions about when to terminate a dialogue, and the user's ability to internally clarify his or her own needs and to choose between competing alternatives:

- **Effects on user confidence:** A classic issue for expert systems, identified early in expert systems research, is the decrease in user confidence in a system—regardless of the quality of its conclusions—if the system appears to “lose focus” during its interaction with the user [6]. Thus given two systems with equal efficiency and precision, we expect user confidence to be higher if the system presents cases which converge consistently towards the final candidate list. As a result, although rank quality does not directly measure confidence, rank quality considerations may be a useful supplement to confidence approaches which focus on assessing the quality of the final result (e.g., [7]).
- **Effects on choosing between alternatives:** Research on expert systems for medical diagnosis showed that diagnostic decision-making may consider not only which diagnosis appears most likely, but also the competition between alternative diagnoses. If the top and next diagnoses are similarly ranked, additional *differential diagnosis* may be needed [8]. Consequently, it may be valuable not only to find a highly-ranked case, but to find a set of best cases to be available for comparison—i.e., for the system to provide a list of top cases early, in order to initiate extra tests to find the values of distinguishing attributes.
- **Effects on the user's ability to identify needs:** Precision and efficiency focus on the ability to drill down to a single case relevant to a fixed problem description. This conception is apt for traditional CCBR troubleshooting tasks. However, as observed in work by McSherry [9] and by McCarthy et al. [10], for newer CCBR areas such as shopping recommenders, the user may initially provide information that is inaccurate or needs to be revised in order to retrieve the right case—the dialogue itself may change the target of the retrieval. Providing the user with a case list with high rank quality early on gives the user an early idea of alternatives consistent with early attribute choices, enabling changing parameters for a new search if those alternatives are not satisfactory.

When comparing rank quality to precision and efficiency, a natural question is whether variants of precision and efficiency could be measured incrementally, by using simulated dialogues in which the best case is “selected” at each step, and precision and efficiency calculated accordingly. However, efficiency calculated in this way would be uninformative, for the incremental efficiency measure would merely be a count of the number of questions asked. Incremental precision values could be more meaningful, but ultimately, a user model is still required to select a case at each step, and we will show that this user model can have a strong impact on experimental results. Thus considering rank quality has benefits even compared to incremental precision criteria.

4 Rank Quality Considerations and Formal Definition

Although rank quality is intuitively easy to grasp, to develop a suitable comparison function is surprisingly subtle. Due to space limitations, we cannot discuss this fully

here, but we illustrate a few issues. Recall that the basic task is to compare the k top-ranked cases in the candidate and ideal lists. Issues include:

- **Handling ties between cases on the lists:** When multiple cases are equally similar to the target (as may be more likely when not all information is available, blurring distinctions), the comparison function must break ties to obtain a linear ordering of candidate cases. This process can have a strong effect on rank quality.
- **Handling boundary splits:** The set of tied cases may extend past the boundary of the list of k top cases presented to the user. The placement of some tied cases outside the boundary could distort results. This requires methods that are not unduly influenced by sequences of ties extending beyond the boundary.
- **Avoiding undue influence from list length:** One possible approach to handling the boundary split problem would be to use a threshold-based retrieval criterion instead of k NN. However, for the measures we considered, longer lists tended to be scored better than short ones, suggesting that it is desirable to avoid comparison of lists with dramatically different lengths.

Our list comparison function calculates the difference in the weighted sum of distances for both lists, with distances weighted by rank. Ties in the candidate list are handled by applying, to all cases in the tied sequence, the average of the weights w_m through w_n of the cases in that sequence. In this way, the arbitrary ordering of the tied cases is irrelevant in the weighted sum, as the same weight is applied to all. Thus all cases in the sequence have the same effect. Splits across the k -boundary in the candidate list are handled by a slight *expansion* or *contraction* of the candidate list from length k to length \hat{k} , to either include or exclude the entire sequence which was originally split. The decision of expansion or contraction depends on which would result in the smaller change in list length. More formally, for weights w_j set as explained below, we define:

$$c(L_t, L_{\hat{t}}) = \begin{cases} 0 & \text{if } \hat{k} = 0 \\ 1 - \frac{\sum_{i=0}^{\hat{k}-1} \hat{w}_i \text{distance}(t, L_{\hat{t}}[i]) - \sum_{i=0}^{k-1} w_i \text{distance}(t, L_t[i])}{\sum_{i=0}^{\hat{k}-1} w_i} & \text{otherwise} \end{cases}$$

$$\hat{w}_i = \begin{cases} w_i & L_{\hat{t}}[i] \text{ is not involved in a tie} \\ \frac{\sum_{j=m}^n w_j}{(n-m+1)} & \text{otherwise} \end{cases}$$

$$\hat{k} = \begin{cases} k & \text{startIndex} = \text{endIndex} \quad ; \text{ no splitting} \\ \text{startIndex} & k - \text{startIndex} < (\text{endIndex} - \text{startIndex} + 1)/2 \\ \text{endIndex} + 1 & \text{otherwise} \end{cases}$$

where *startIndex* and *endIndex* are the 0-based indices marking the start and end of the sequence of tied cases splitting across the boundary. Note that the contraction process assures that the denominator of the previous weight formula is always nonzero.

Weight Assignment: Exponentially decreasing weights emphasize higher-ranked cases:

$$w_i = \text{min}V + (\text{max}V - \text{min}V) \left(\frac{i - (k - 1)}{k - 1} \right)^{2\lambda}$$

λ is a positive integer representing the rate of decrease in the weights, and $minV$ and $maxV$ are the desired minimum and maximum weights, respectively. For the experiments of this paper, $minV = 0$, $maxV = 1$, and $\lambda = 2$. The weights could also be set based on characteristics of how the user examines the case list, if that information were available (e.g., if some user examined all cases in the list equally without regard for their ranking, equal weights would be more appropriate).

Contraction to 0 and Distance Granularity: According to the above formula, when $\hat{k} = 0$, $c(L_t, L_i) = 0$ as well. This situation occurs when all cases in the candidate list, plus a proportionately large number beyond the list, are equally similar to the target problem. For example, if $k = 10$ and the top 25 retrieved cases are tied, then this list is contracted to exclude the tied cases (reflecting that the majority of tied cases were excluded even before the list contraction). This results in $\hat{k} = 0$, for a rank quality of 0. We call this *contraction to 0*. This process is consistent with the intuition that rank quality should be low when the system’s ordering of candidate cases is arbitrary, with no grounds for distinguishing any candidate cases from many non-candidates.

We note, however, that the result may be counterintuitive in a special case. For example, if 10 cases are presented and the *ideal* list has the top 25 cases tied, and the candidate list presents 10 of these cases, its rank quality would still be 0, even though intuitively no alternative list would be better. Thus in this case, the function reflects not only objective case suggestion quality but the system’s ability to select which cases to present. Although we are exploring alternatives to reflect objective rank quality alone, fully capturing intuitions in such a function has proven surprisingly difficult, with “natural” alternatives having more severe problems. From our own observations, contraction to 0 appears very unlikely to happen in standard domains with non-categorical attributes, though it may occur in domains with few attributes (proportional to case base size), if all of them are categorical. To quantify the extent to which this might cause difficulties for a leave-one-out test, we can determine the likelihood of tests avoiding contraction to 0 by calculating the *distance granularity*—the average proportion, over the cases, of unique distances in the case base. For each case c_j , define $uniqueDCount(c_j)$ as the number of unique values of $distance(c_j, c_i)$, for all c_i in case base CB . Then:

$$caseGranularity(c_j, CB) = uniqueDCount(c_j)/|CB|$$

$$distanceGranularity(CB) = \frac{\sum_i caseGranularity(c_i, CB)}{|CB|}$$

5 Experimental Comparisons of the Measures

We conducted experiments to explore the sensitivity of precision–efficiency approaches to the case selection strategy (the simulated user), and to examine the information provided by the different measures. All experiments were conducted using the Indiana University Case-Based Reasoning Framework (IUCBRF), a freely-available open-source Java framework for rapid and modular CBR system development [11]. All datasets are from the University of California-Irvine (UCI) repository [4], for classification. These

experiments use the Pima dataset with entirely numerical attributes, and the Spect and Zoo datasets with entirely categorical attributes.

We apply the measures to evaluating CCBP performance for systems using different similarity assessment strategies for partially-specified cases [3]. Under a representativeness assumption, the case base is used to predict information about as-yet-unasked attributes in a problem. These strategies are selected here not to be evaluated per se, but to illustrate the measures of this paper. The strategies behave as follows when comparing corresponding attribute values for which at least one is unknown:²

- **DefaultDifference(0)** (DD) - Assume a difference of 0 between the attribute values.
- **FullAggregate** (FA) - Assume the unknown value is the aggregate (e.g. mean for numeric attributes, majority vote for categorical) value of that attribute in the entire case base.
- **NNAggregate(DefaultDifference(0))** (ND) - Assume the unknown value is the aggregate attribute value of the nearest cases. “Nearest” is defined by a similarity measure using DefaultDifference(0) to handle missing attributes.
- **NNAggregate(FullAggregate)** (NF) - Similar to above, except “nearest” is defined by a similarity measure using FullAggregate to handle missing attributes.
- **RegionAggregate(DefaultDifference(0))** (RA) - Assume the unknown value is the aggregate attribute value of cases in the corresponding region. The regions are pre-determined offline by a similarity measure using DefaultDifference(0) to handle missing attributes.

5.1 Experimental Setup

This experiment generally follows the template laid out in [2]. Five systems were constructed, one for each of the five missing attribute strategies above. A case is chosen and its attributes are gradually “revealed” as answers to system questions. (Note that we have done leave-one-out rather than leave-one-in; for the datasets used here, which are not irreducible, we expect comparable conclusions with either approach).

Questions are selected randomly by the system, in order to remove the effects of particular question selection strategies. Each target is tested multiple times, for different random question patterns, with the results averaged. Each time a question is answered, the system calculates the rank quality of the candidate list obtained according to its missing attribute strategy. This process continues until every question is answered, regardless of when a final case is selected for the purpose of precision and efficiency calculations.

Simulated user design: To calculate precision and efficiency first requires tracing the dialogue until it reaches a stopping point — when the user selects a case. We informally describe a *restrictive* simulated user as one with criteria for case selection that are more difficult for a candidate list to meet. Four types of simulated users were examined:

² The 2004 paper uses slightly different names, as follows: FullMean instead of FullAggregate; NNMean instead of NNAggregate; and RegionMean instead of RegionAggregate.

- **T5** (top 5) - Select a case when it is among the top 5 candidate cases and is below a distance threshold h . If multiple cases fall below the threshold at the same time, the highest ranking one is selected.
- **T1** (top 1) - Select a case when it is the top-ranked candidate case and is below a distance threshold h .
- **A5** (average 5) - When the average distance of the top 5 candidate cases is below a distance threshold h , randomly select a case with a distance less than the average (that is, one of the better cases of the candidate list).
- **DL** (dialogue length) - Select a case when it is the top-ranked candidate case and it contains the target solution. The efficiency for this user is the *dialogue length*. The precision will always be 1.0 except for rare circumstances where no case is selected and the top case at the end of a dialogue does not contain the target solution.

Precision and Efficiency Calculation: Efficiency is $1 - \frac{\text{revealed}}{\text{total}}$ for *revealed* the number of attributes revealed and *total* the total number of attributes in the domain (thus efficiency is between 0 and 1). Precision is 1 if the selected case solution is identical to the target case solution, 0 otherwise. Note that only one precision and efficiency measurement is taken per dialogue, upon case selection.

Threshold calculation: Each of these users depends on a distance threshold h . We chose to form a “level playing field” for comparison by calculating thresholds for each domain as follows. A leave-one-out process is used to calculate a set of thresholds for a range of restrictiveness, controlled by the choice of a parameter $b \in (0, 1)$ reflecting the proportion of the case base to contain in a given neighborhood. For $|CB|$ the size of the case base, we set $i = \text{round}(b \cdot |CB|)$. For each case, cases are retrieved using the problem of the selected case as the query. The distance between the target and the i th-ranked case for each retrieval is averaged across the case base (that is, throughout the leave-one-out process), and is set as the fixed distance threshold for case selection for the corresponding domain. In this way, domain-specific influences are accounted for, and the threshold is chosen based on its effect in relation to the domain properties. Four thresholds were computed for each domain, from fairly restrictive to very unrestrictive thresholds, corresponding to b values of 0.05, 0.10, 0.15, and 0.30.

Distance granularities: Distance granularities are computed for the three domains of this experiment. Pima is an entirely non-categorical domain, and so it is not surprising that its distance granularity is 0.999. Zoo, with all categorical attributes, has a much lower distance granularity of 0.116, though no contractions to 0 occurred in our experiments. Spect, also with all categorical attributes, has a distance granularity of 0.064. This proved low enough for some inappropriate contractions to 0 to occur, but the general patterns and conclusions as seen in the other domains remain, showing that even in this situation, the comparison function provided useful information.

5.2 Results and Discussion

Preliminary Notes: Fig. 1 shows precision and efficiency results for various users, and Fig. 2 shows selected rank quality results. The Pima rank quality results (not shown due

Domain	A5, 0.05	T1, 0.05	T5, 0.05	T5, 0.10	T5, 0.15	T5, 0.30	DL
Zoo	RA 0.401	RA 0.394	DD 0.415	DD 0.431	DD 0.425	FA 0.456	RA 0.413
	ND 0.386	ND 0.375	ND 0.400	ND 0.418	ND 0.415	NF 0.453	ND 0.409
	DD 0.315	DD 0.315	RA 0.396	RA 0.412	RA 0.412	DD 0.447	DD 0.393
	NF 0.277	NF 0.305	NF 0.370	NF 0.386	NF 0.380	RA 0.440	NF 0.339
	FA 0.199	FA 0.273	FA 0.341	FA 0.363	FA 0.367	ND 0.444	FA 0.293
	rng 0.202	rng 0.121	rng 0.074	rng 0.068	rng 0.058	rng 0.012	rng 0.120
	Spect	NF 0.475	FA 0.470	DD 0.422	DD 0.491	DD 0.496	DD 0.498
FA 0.474		NF 0.469	ND 0.412	ND 0.484	ND 0.491	ND 0.495	ND 0.486
DD 0.459		DD 0.467	RA 0.411	RA 0.483	RA 0.490	FA 0.493	DD 0.446
RA 0.458		RA 0.458	FA 0.380	NF 0.482	FA 0.490	RA 0.493	FA 0.399
ND 0.458		ND 0.452	NF 0.379	FA 0.481	NF 0.490	NF 0.493	NF 0.397
rng 0.017		rng 0.018	rng 0.043	rng 0.010	rng 0.006	rng 0.005	rng 0.106
Pima		RA 0.301	RA 0.309	FA 0.422	DD 0.406	DD 0.419	DD 0.435
	ND 0.274	ND 0.286	NF 0.421	FA 0.358	FA 0.389	FA 0.423	ND 0.485
	NF 0.246	NF 0.266	ND 0.415	NF 0.358	NF 0.373	NF 0.416	RA 0.482
	FA 0.223	DD 0.249	DD 0.387	RA 0.352	RA 0.362	ND 0.392	NF 0.452
	DD 0.201	FA 0.243	RA 0.358	ND 0.344	ND 0.360	RA 0.378	FA 0.441
	rng 0.100	rng 0.066	rng 0.064	rng 0.062	rng 0.059	rng 0.057	rng 0.044

(a) Efficiency

Domain	A5, 0.05	T1, 0.05	T5, 0.05	T5, 0.10	T5, 0.15	T5, 0.30	DL
Zoo	RA 0.997	RA 0.995	RA 0.996	RA 0.989	RA 0.987	RA 0.881	ND 1.000
	ND 0.997	ND 0.995	DD 0.992	ND 0.981	ND 0.982	ND 0.879	FA 0.999
	DD 0.996	DD 0.991	ND 0.991	DD 0.971	DD 0.968	DD 0.852	DD 0.999
	NF 0.996	NF 0.975	FA 0.981	FA 0.956	FA 0.954	NF 0.753	RA 0.999
	FA 0.995	FA 0.973	NF 0.980	NF 0.956	NF 0.946	FA 0.700	NF 0.999
	rng 0.002	rng 0.022	rng 0.016	rng 0.033	rng 0.041	rng 0.181	rng 0.001
	Spect	RA 0.674	ND 0.637	DD 0.673	RA 0.598	RA 0.629	RA 0.602
ND 0.665		FA 0.632	RA 0.660	ND 0.597	ND 0.601	ND 0.595	ND 0.999
FA 0.641		NF 0.631	ND 0.659	DD 0.579	DD 0.581	DD 0.586	DD 0.994
NF 0.641		RA 0.627	NF 0.628	FA 0.576	FA 0.569	NF 0.549	FA 0.986
DD 0.633		DD 0.596	FA 0.617	NF 0.560	NF 0.559	FA 0.544	NF 0.986
rng 0.041		rng 0.042	rng 0.056	rng 0.038	rng 0.070	rng 0.058	rng 0.013
Pima		RA 0.666	RA 0.660	ND 0.706	RA 0.648	RA 0.648	RA 0.644
	DD 0.634	DD 0.591	RA 0.706	DD 0.592	DD 0.591	DD 0.580	NF 0.998
	ND 0.632	ND 0.581	DD 0.693	ND 0.565	ND 0.575	ND 0.562	DD 0.996
	FA 0.629	NF 0.576	NF 0.650	NF 0.554	NF 0.553	NF 0.551	RA 0.995
	NF 0.629	FA 0.561	FA 0.606	FA 0.531	FA 0.522	FA 0.530	ND 0.994
	rng 0.037	rng 0.099	rng 0.100	rng 0.117	rng 0.126	rng 0.114	rng 0.004

(b) Precision

Fig. 1. a) Efficiency and b) precision results for various users and domains. Each column represents a case selection strategy, with column heading showing user type and b value with which to compute the threshold. Columns are organized from most restrictive user on the far left, to least restrictive on the far right, except for DL, which has varying restrictiveness dependent upon the number of cases in the case base containing a given target solution. Each cell lists the five missing attribute strategies in order of decreasing performance. The final number in each cell (“rng”) indicates the range of values in that cell.

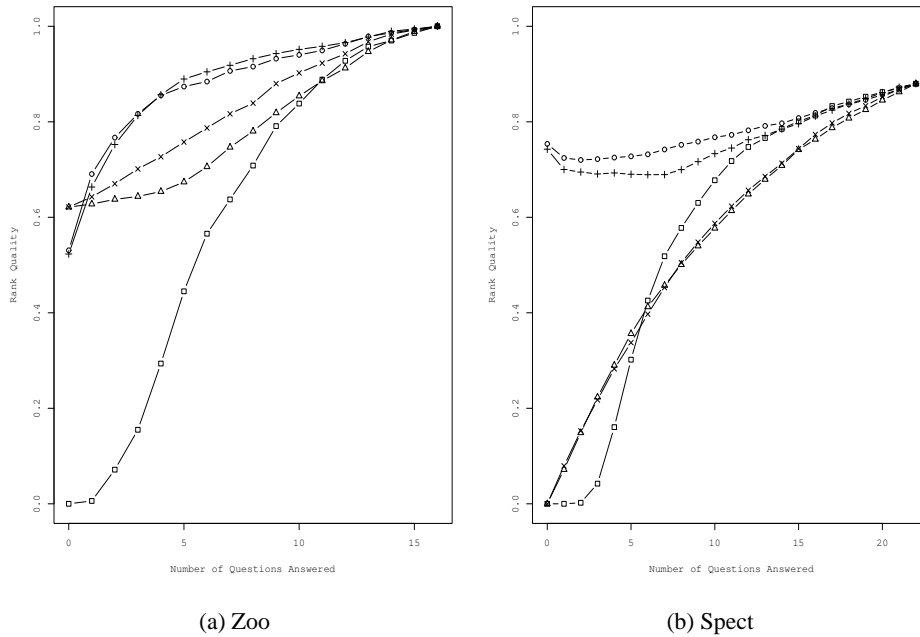


Fig. 2. Rank quality results for the a) Zoo, and b) Spect, domains. The lines correspond to the strategies as follows: RA \circ , ND $+$, DD \square , NF \times , and FA \triangle .

to space limitations) are similar to the Spect and Zoo results in overall suggestions of missing attribute strategies, although the strategies were not as distinguished in Pima as in the other domains, and their order did not change for different numbers of attributes. Note that the strategies vary in performance across domains, both in magnitude, and occasionally in overall ranking. This is not interpreted as evidence of problems with precision and efficiency, and we note that similar domain differences are evident for rank quality.

We also caution the reader not to expect an exact connection between rank quality and precision–efficiency. Precision and efficiency are performance snapshots at the *end of the dialogue*, which changes depending on when a case is selected. Consequently, these measures are not easily comparable to any single portion of a rank quality graph, although to some extent ranges can be compared, as shall be demonstrated below.

Also note that precision measures solution applicability, and efficiency measures speed of case selection, while rank quality is a measure of similarity. Clearly there is a connection, in that a similar case is assumed to have an applicable solution, and a candidate list with high similarity will likely be selected from sooner. Nevertheless, these are different measures, with solution applicability depending not only on the similarity but also on less definable domain properties, and with case selection also depending on user properties. Thus, efficiency and precision measures are not directly comparable to rank quality, though there are connections in the general trends shown by both.

Effects of Different User Models on Precision and Efficiency The first experiment explores the potential sensitivity of precision–efficiency approaches to the chosen user model. Fig. 1 presents efficiency and precision results, organized to place more restrictive case selection criteria towards the left, with restrictiveness decreasing towards the right. (The one exception is DL, with varying restrictiveness dependent upon the number of cases in the case base containing a given target solution.) Each cell orders the performance of each strategy according to the measure of the table. The bottom number in each cell represents the range of values in that cell.

In some cases, performance is quite similar across similarity assessment strategies. When there are clear distinctions between the strategies’ performance, RA, ND, and DD tend to be the best by both measures, with NF and FA tending to be the worst. This supports a reasonable regularity of conclusions across user types. This is a good result for the utility of precision and efficiency, for if they were totally dependent on user type, then generally applicable conclusions would be very difficult to make.

However, the regularity of conclusions is far from absolute. For example, consider the efficiency results for the Zoo and Spect domains in Fig. 1(a). The range of values decreases given decreasing restrictiveness. With Spect, for T5 with higher thresholds, the range of values is very low. If only these users were examined, a researcher might erroneously conclude that there is no significant distinction in missing attribute strategy performance. Another example is Pima’s precision results, which, surprisingly, were higher for (T5, $b = 0.05$) than for more restrictive users. The reasons for this are unclear. Nevertheless, these examples demonstrate that the user model affects conclusions about system performance.

Other efficiency examples demonstrate this same point. Considering the efficiency performance of RA and DD for the Zoo and Pima domains, we note that comparatively, RA is favored for the most restrictive users, but is closer to the middle or even the bottom for the less restrictive users. On the other hand, the ranking of DD increases for less restrictive users. Thus the researcher’s choice of RA and DD may depend on the type of user examined. This difference in performance for different users is an important result, but is captured in efficiency measures only by careful and broad user modeling.

The precision results in Fig. 1(b) reflect this issue as well. It is interesting to note the relationship between restrictiveness and range of precision values. All three domains demonstrate the opposite relationship to that observed for efficiency: Although the range of efficiency values for a given strategy decreases given *decreasing* user restrictiveness, range of precision decreases given *increasing* user restrictiveness.

Another interesting result is that in some domains, results appear to be fairly independent of the user model. For the Zoo domain, for almost every user, the precision results are quite close, and extremely high. It appears that in this domain, the relationship between problem similarity and solution applicability is very strong. Only for a very unrestrictive user model can clear distinctions be seen.

As discussed previously, if it is possible to select a user model known to capture specific user characteristics, the sensitivity of precision–efficiency judgments to the user model is not an issue, and is in fact desirable. However, we are unaware of human-subjects studies in the literature evaluating such models. For many domains, it may be unclear which models fit, and developing the right models may be impractical, or even

impossible. In that case, the measures' sensitivity to possibly arbitrary characteristics of simulated users is problematic. This supports the appeal of approaches such as rank quality, which do not depend on case selection criteria in a user model.

Revealing Trends in the Three Measures: As discussed above, there are limits to the comparability of the measures, but this section shows that the rank quality results are comparable to precision–efficiency in a broad sense, and the ability to apply rank quality at any point in the dialogue can reveal performance trends crucial to system analysis.

The most obvious way to compare precision and efficiency to rank quality is to examine the final conclusions which can be drawn from each. According to the efficiency and precision results in Fig. 1, the best strategies are typically RA and ND, and often (but not always) DD. The worst strategies are generally NF and FA. Similar results for rank quality are reflected in Fig. 2. In these graphs, RA and ND are generally the best whenever there is a clear winner, with NF and FA among the worst.

One of the most interesting comparisons comes from the DD results in the Spect domain. Here DD efficiency is approximately 0.5 for less-restrictive users (T5, $b = 0.30, 0.15, 0.10$), corresponding to an average of 11 questions answered in a dialogue. For the same users, precision for DD is approaching that of ND and RA, with the overall ordering being RA, ND, DD, FA/NF (roughly tied). This corresponds to the same ordering in the rank quality graph, for 11 questions answered: RA, ND, DD, FA/NF.

The efficiency of DD for T5, $b = 0.05$, a model of a more restrictive user, decreases to 0.422, corresponding to an average of 12.1 questions answered. The precision measure for this user for DD is 0.673, slightly better than RA and ND. Although rank quality results for 12.1 questions answered still show DD lower than RA and ND, DD is in fact rising at a more rapid pace, and is closer to them than for 11 questions answered.

This trend for DD continues for the T1 and A5 users in the Spect domain. Why these users' efficiency results differ from those of more restrictive users is a subject for future study. The efficiency results for DD are higher than the T5, $b = 0.05$ efficiency results, but lower than the other T5 efficiency results. Specifically, the average efficiency between the two users T1 and A5, 0.463, corresponds to approximately 11.8 questions answered. Upon examination of the rank quality graph, we would therefore expect the corresponding DD precision results to be lower than the T5, $b = 0.05$ results, and higher than the other T5 results, as observed. Thus again we see a general correspondence between the rank quality and precision and efficiency measures.

Again, we do not expect an exact correspondence between efficiency and precision and rank quality, given the clear differences in their design. However, the correspondence among general conclusions is reassuring in that rank quality captures the broad outlines of the more traditional measures, while providing much more information. Although precision and efficiency provide a single snapshot of performance of a single case upon selection, rank quality can be used to show the development of the candidate list across the entire dialogue. The rank quality graphs illustrate that DD starts out very poorly, but rises quickly until, when most questions have been asked, it performs nearly as well as the other strategies. In fact it can be seen in the rank quality graphs that nearly all strategies perform approximately equally when most questions have been asked. This is intuitive, for missing attribute strategies have less opportunity to distinguish themselves when there are fewer missing attributes. On the other hand, when

very few questions have been asked in the dialogue, rank quality shows that RA and ND still consistently perform fairly well. FA and NF fare well in the Zoo domain at this stage, but not in Spect. DD, on the other hand, is consistently bad at this stage. Such conclusions are readily apparent in viewing rank quality graphs, but would be much more difficult to make by examining precision and efficiency results, hoping to select the right set of users to get a useful range of data.

6 Related Work

Many existing CCBR evaluation efforts use forms of efficiency or precision for performance evaluation, with some approaches relating to rank quality.

Variations of Efficiency, Precision, and Rank Quality: Evaluation criteria related to efficiency are widely used. For example, McCarthy et al. [10] examine the process of *dynamic critiquing*, in which the user refines the problem description in reaction to the presented candidate list. The number of “tweaks” performed by the user is measured, similar to efficiency. Precision does not apply in the same manner, because presumably the user does not stop critiquing until a satisfactory result is obtained. McSherry [12] uses the leave-one-in process in the context of *irreducible* case bases, in which each case has a unique solution, and there is at most one applicable case for any given problem. McSherry suggests a *recall* and slightly modified precision measure for evaluation of CCBR systems of this nature. Recall is the percentage of queries in which the single perfect case is among the retrieved cases. His precision measure captures the probability that the single applicable case could be selected at random from the candidate list. In [13], the target case of leave-one-in serves as the single applicable case, the goal to be ultimately selected. *Conversational efficiency* is then defined as the number of questions required to get a 100% similarity rating with this target. This is similar to an efficiency measure with a case selection threshold of 100%.

The *retrieval accuracy* measure of Gupta et al. [13] is reminiscent of both precision and rank quality, although with key differences. Retrieval accuracy measures the average rank of the applicable case. It is similar to precision in that it considers applicability of a single case, and to rank quality in that it examines the rank of a case throughout the dialogue. However, this measure does not consider the ranks of the other cases, which may be important for the reasons discussed in Section 3.

In [14], the frequency of successful retrievals is measured, where a successful retrieval has the top three cases of the ideal list in the top five of the candidate list. Thus, similar to rank quality, it compares the candidate list to the ideal, though less information about the full list is gathered.

Measures for Other CCBR Properties: There are a number of measures used in evaluation of CCBR systems that fall outside the scope of the three measures considered here. For example, McSherry [9] measures the length of explanations of retrieval failures, and how many compromises to the original query are required for recovery from these failures. Gupta et al. [13] present three measures related to question ranking: conversational accuracy (how suitable the question rankings are), the number of questions

presented to the user at a time, and conversational adaptiveness (the ability of the system to adapt the dialogue to the user's ability level). The same paper also presents two measurements aimed at knowledge engineering: the effort required to insert a case or a new attribute type into the case base.

Connections beyond CBR: The broad question of how to evaluate the quality of a list of provided information resources is also of great importance outside of CBR, for tasks such as ranking Web search results. Although we are not aware of any directly-applicable results from that literature, steps taken there have some bearing on rank quality research for CCBR. It is a common practice in Web search research to use humans as judges of the relevance of a retrieved site. However, in [15] it is argued that human relevance judgments do not lead to stable measures, and that disagreements about even the single most relevant result are frequent. If this holds true for CCBR, it would be an additional impediment to relying on human ranking judgments. If users are presented with search results with detailed summaries, and followed links are automatically marked "relevant," it is possible to use this information to estimate relevance during search engine use [16], a method which might be usable for CCBR. Various approaches exist for aggregating relevance ratings to obtain a rating of a list of search results, including the *ranked half life* measure, which calculates the degree to which relevant documents are located at the top of the list [17], and a measure of the expected number of irrelevant documents to be searched through before relevant documents are found [18].

7 Conclusion

This paper has examined issues in CCBR system evaluation and has proposed rank quality for CCBR system evaluation. This approach enables evaluation at any point in the dialogue, and removes the requirement, needed by precision and efficiency methods, of modeling the user's case selection decision. The paper has presented experimental results illustrating the value of decreasing dependence on a case selection model, by demonstrating that—although the case selection model must often be selected somewhat arbitrarily in practice—it may strongly influence evaluation results. Rank quality can provide useful information during the CCBR conversation, to help select strategies which provide the user with useful cases early on. This may be valuable for increasing user confidence, helping the user to choose between alternatives, and identifying needs for CCBR tasks such as supporting product recommendation. Due to the surprisingly subtle issues involved in rank quality calculation, and the fundamental importance of CCBR to CBR applications, we see further exploration of such criteria and their relationship to user satisfaction as a promising area for future research.

8 Acknowledgments

We thank the anonymous ECCBR reviewers for their very valuable comments.

References

1. Watson, I.: *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann, San Mateo, CA (1997)
2. Aha, D., Breslow, L.: Refining conversational case libraries. In: *Proceedings of the Second International Conference on Case-Based Reasoning*, Berlin, Springer Verlag (1997) 267–278
3. Bogaerts, S., Leake, D.: Facilitating CBR for incompletely-described cases: Distance metrics for partial problem descriptions. In Funk, P., Calero, P.A.G., eds.: *Proceedings of the Seventh European Conference On Case-Based Reasoning*, Berlin, Springer (2004) 62–76
4. Newman, D., Hettich, S., Blake, C., Merz, C.: *UCI repository of machine learning databases* (1998)
5. McSherry, D.: Minimizing dialog length in interactive case-based reasoning. In: *Proceedings of the seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, San Mateo, Morgan Kaufmann (2001) 993–998
6. Buchanan, B., Shortliffe, E.: *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, MA (1984)
7. Cheetham, W., Price, J.: Measures of solution accuracy in case-based reasoning systems. In: *Proceedings of the Seventh European Conference On Case-Based Reasoning*. (2004) 106–118
8. Miller, R., Pople, H., Meyers, J.: Internist-i, an experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine* **307**(8) (1982) 468–476
9. McSherry, D.: Incremental relaxation of unsuccessful queries. In Funk, P., Calero, P.G., eds.: *Proceedings of the Seventh European Conference on Case-Based Reasoning*. Volume 3155., Berlin, Springer-Verlag (2004) 331–345
10. McCarthy, K., Reilly, J., McGinty, L., Smyth, B.: Experiments in dynamic critiquing. In: *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, New York, NY, USA, ACM Press (2005) 175–182
11. Bogaerts, S., Leake, D.: IUCBRF: A framework for rapid and modular CBR system development. Technical Report TR 617, Computer Science Department, Indiana University, Bloomington, IN (2005)
12. McSherry, D.: Precision and recall in interactive case-based reasoning. In: *ICCBR '01: Proceedings of the 4th International Conference on Case-Based Reasoning*, London, UK, Springer-Verlag (2001) 392–406
13. Gupta, K.M., Aha, D.W., Sandhu, N.: Exploiting taxonomic and causal relations in conversational case retrieval. In: *ECCBR '02: Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, London, UK, Springer-Verlag (2002) 133–147
14. Kohlmaier, A., Schmitt, S., Bergmann, R.: Evaluation of a similarity-based approach to customer-adaptive electronic sales dialogs (2001)
15. Voorhees, E.M.: Evaluation by highly relevant documents. In: *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, ACM Press (2001) 74–82
16. Boyan, J., Freitag, D., Joachims, T.: A machine learning architecture for optimizing web search engines. In: *Proceedings of the AAAI Workshop on Internet-Based Information Systems*, Portland, Oregon (1996)
17. Borlund, P., Ingwersen, P.: Measures of relative relevance and ranked half-life: performance indicators for interactive ir. In: *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, ACM Press (1998) 324–331
18. Cooper, W.S.: Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation* **19**(1) (1968) 30–42