Evaluation of Textual CBR Approaches

Stefanie Brüninghaus and **Kevin D. Ashley**
Intelligent Systems Program and Learning Research and Development Center
University of Pittsburgh, Pittsburgh, PA 15260
steffi+@pitt.edu, ashley+@pitt.edu

Importance of Evaluation and Issues for Textual CBR

Evaluation is a crucial step in a research project, it demonstrates how well the chosen approach and the implemented techniques work, and can uncover limitations as well as point toward improvements and future research. A formal evaluation also facilitates comparing the project to previous work, and enables other researchers to assess its usefulness to their problems. Evaluating Textual CBR system is particularly challenging: The system will often comprise multiple aspects, e.g., interface, index-assignment, or case-based reasoning proper. The evaluation has to be carefully designed so that it disentangles the performance of those individual components which are to be the focus of the research project, or captures the performance of the overall system in solving a practical problem in industry. Textual CBR incorporates aspects of various disciplines which have different views on how to measure performance, and it is arguable which one of these measures is the best choice. In general, for Textual CBR, how to evaluate is still mostly an open issue (CBRIR 1998), and unfortunately, in some systems not even addressed at all. Here, we will discuss three major issues with respect to the evaluation of Textual CBR systems:

1. *What performance measure should be chosen?* The widely-used IR measures precision/recall do not capture all of the information required for assessing the performance of a Textual CBR system.
2. *What should the performance be compared to?* In order to allow an assessment of the system, evaluation requires a standard of comparison, which is often difficult to define for Textual CBR systems.
3. *What to focus the evaluation on?* It is often difficult to define what is being evaluated, as Textual CBR systems comprise a number of different aspects, and are often an integral part of an overall system.

There is no single answer to any of these questions, and the design of a formative evaluation strongly depends on the task at hand. For a retrieval-oriented system different aspects matter than for a system where further reasoning with or about the textual cases will be performed. For reasons presented elsewhere in this volume (Brüninghaus & Ashley 1998) our focus will on the latter type of system, and we will discuss the decisions we made in evaluating a Textual CBR system.

What Performance Measure Should be Chosen?

Existing Evaluation Measures

In the course of an evaluation, the performance of the system in the experiments has to be recorded. In the definition of the various measures below, we refer to the number of test instances as N, the other variables are used as defined by the following contingency table:

	Relevant	Not Relevant
Retrieved	*TP* True Positives	*FP* False Positives
Not Retrieved	*FN* False Negatives	*TN* True Negatives

In IR, the most widely used measures are precision and recall. Precision captures the portion of retrieved objects relevant to the user's query, while recall measures what portion of objects relevant to the user's query was actually retrieved by the system. Unfortunately, in recent work in machine learning, precision and recall have been referred to as accuracy and coverage, which shows that it is imperative explicitly to define the measures used, in order to avoid misinterpretations. Formally, the measures are defined as:

$$\text{Precision:} \quad p = \frac{\text{Relevant and Retrieved}}{\text{Retrieved}} = \frac{TP}{TP + FP}$$

$$\text{Recall:} \quad r = \frac{\text{Relevant and Retrieved}}{\text{Relevant}}$$

$$= \frac{TP}{TP + FN}$$

These definitions of the measures were sufficient and appropriate for the early IR systems, which were merely capable of boolean search. In those systems, the user expressed his query as a boolean combination of keywords, and the systems retrieved the documents that fulfill constraints represented by the query. Today's IR systems usually do not assign the label relevant/not relevant. Vector-space and probabilistic models return a linear list of documents, ordered by their (estimated) similarity and relevance to the query. Depending on the system-design, the user will have the option to inspect all documents, or a reasonable cut-off point (threshold or count-based) will be chosen. With these systems, the choice of the cut-off point will greatly influence what precision/recall the system achieves: if more documents are returned, more of the relevant documents are being retrieved, and recall will increase. At the same time, more irrelevant documents will also be presented to the user, and precision will decrease. In order to take this trade-off into account, most systems are evaluated by taking the average precision at 0, 10, 20, ... 100 % recall level, and by plotting the curve of precision at those chosen recall points. Alternatively, to capture the information in a single number, the Break-Even-Point is defined the point where recall equals precision. The F-measure weights precision and recall, and is calculated as:

$$F_\beta = \frac{(\beta^2 + 1)pr}{\beta^2 p + r)}$$

The advantage of the F-measure is that it allows to weight a user's preferences between precision and recall, and expresses performance in a single statistic. However, the resulting number can not be easily interpreted, and is mainly of academic interest. Thus, we think that in spite of its advantages, the F-measure as well as the Break-Even-Point are not appropriate for evaluating Textual CBR systems.

In most other fields, the main criterion for a system's performance is how often the system is "making the right decision", without distinguishing between what kind of errors the system makes. We refer to the portion of correct decisions of the system as accuracy, and it is defined as follows:

$$\text{Accuracy:} a = \frac{TP + TN}{N} = \frac{TP + TN}{TP + TN + FP + FN}$$

In a variation of accuracy, a loss measure can be used (Lewis 1995). Depending on the application or the user's preferences, false positives and false negatives may not be weighted equally. If it is crucial to correctly identify every instance of a concept, errors of omission affect system performance more, while in a setting where inaccurately marked positive instances are tolerable, as long as not negative instances slip in, errors of commission are more costly. Similarly, the correct classification of positive and negative instances may have different value to a user. This can be accounted for by assigning subjective weights $\lambda_{TP}, \lambda_{TN}, \lambda_{FP}$, and λ_{FN} to the respective counts, and calculate the loss as:

$$\text{Loss: } l = \frac{\lambda_{TP} TP + \lambda_{TN} TN + \lambda_{FP} FP + \lambda_{FN} FN}{N}$$

This measure makes inter-system comparison more difficult, but is preferable for the evaluation of more task-oriented systems.

If the focus is on the overall project into which textual CBR is integrated, evaluation can also employ user-centered, qualitative measures like profitability or user satisfaction. This is particularly true in applications where test-data and relevance judgements are almost impossible to find, making more formal evaluation impossible. In a Textual CBR system, as we suggest in the first part of this position statement, however, performance generally may be assessed with more quantitative performance measures, because they allow for defining a baseline performance, and facilitate the comparison of different approaches.

Discussion and Comparison of Precision/Recall and Accuracy

Both of the most commonly used measures, precision/recall and accuracy have their strengths and weaknesses, in particular in the context of evaluating a Textual CBR system: Precision/recall are a pair of complementary statistics, and unless the entire data are plotted, information is lost. Depending on the application, it can be important that an object is classified as "non-relevant", e.g., in legal argumentation, the absence of a factor can be crucial in distinguishing it. Precision/recall do not (directly) reflect the correct non-retrieval of these instances; they only consider relevant instances. The measures thus can be misleading, in particular when there are many relevant objects. The 11-point level calculations are appropriate for applications where large numbers of documents are processed, and where one sensibly can calculate precision at 11 recall-levels (in IR, a million documents is no longer an unrealistic problem scale). In CBR-related applications, the number of labeled documents is often magnitudes smaller than in IR projects. Also, cross-validation often interferes with precision/recall curves, as among training/test-set splits, relevance scores are usually not comparable. Precision/recall should also be compared to a standard, not taken per se.

On the other hand, accuracy can be very misleading when applied to concepts with very few positive instances. An utterly useless-in-practice, but high-accuracy classifier would always label new documents as "concept does not apply" - it would achieve an impressive accuracy of close to 100

Our Approach

In our experiments (Brüninghaus & Ashley 1997), we made a compromise by taking both evaluation measures into account. We found for our problem accuracy to be the appropriate measure, because our task is not retrieval, but rather classification where the negative instances also matter. However, in addition to accuracy, we interpreted the results in the light of precision/recall.

What should the performance be compared to?

Importance of a Baseline Standard and Comparison Methods

In evaluating a novel approach, it is important to determine whether its performance is better or worse than known approaches. Thus, one has to compare the new system to previously published or reconstructed results or to a well-chosen baseline performance. Arguing for the difficulty of the problem, and presenting experimental results does not prove that trivial approach may have lead to equal performance. Using a standard of comparison enables others to interpret the experimental results. A naked precision/recall curve is of very limited use, since it does not reflect the level of complexity of the concepts to be assigned, and it also does not account for the difficulty or size of the data set.

For textual CBR systems, choosing a standard of comparison is particularly difficult. First, few if any systems can serve as a standard. Second, if the textual CBR is part of a larger overall approach, it can be too difficult to separate its performance, and carry out a comparative evaluation. Under those circumstances, though, it is not clear what the contribution of an approach is, or whether it works at all.

When comparing the performance of two or more systems or approaches, the comparison can be purely qualitative, but can also include a more in-depth statistical analysis. When two or more sets of (potentially corresponding) performance measurements are available, statistical inference methods allow concluding with a chosen certainty whether there are statistically significant data-sets, by using an ANOVA. Paired and unpaired T-tests provide evidence whether one set of performance measures is better than another one, or whether differences are mainly random.

Our approach

In the evaluation of our experiments, we chose a default strategy as the baseline against which we compared the different classification methods. We assumed that a reasonable strategy to achieve high accuracy scores is always to assign ”no”. A good method will have to be powerful enough to find the positive instances, but also not do much worse in not labeling the negative instances as negative. So, we would like an algorithm that is at least as good in terms as accuracy as our default strategy. We did not perform any statistical analysis of the data, however, because we found that the performance of neither algorithm was satisfactory yet, and we thus did not attempt to show that the performance is better than the default strategy.

Use of Test Collections

In machine learning and information retrieval, often standard collections are used to compare the performance of different systems in a generally accessible way. The advantage is that an objective standard for comparison exists. For Textual CBR, however, such a methodology is not applicable. Standardized collections are not appropriate for systems that incorporate domain knowledge. As we have pointed out before, for a Textual CBR system, domain knowledge is an important component which can not be excluded in evaluation.

Moreover, one should not only apply a reasonable standard of comparison in order to show the contribution, but also investigate the underlying data set to determine whether it has some hidden bias and how correct the data are. For instance, the most widely used collection in text classification experiments, the Reuters collection (Yang 1997), has a number of shortcomings, which are often not mentioned explicitly. One major problem with the collection is that as many as 58 % of the documents in the pre-defined training-set do not have any categories assigned, although they probably should . There are multiple versions of this collection, with different training/test-splits, where documents are assigned randomly or by date; the latter potentially creating a bias. Finally, as noted in (Yang 1997), by selecting only a subset of the documents or categories (Koller & Sahami 1997), the results may seem too optimistic.

Thus, we think that for Textual CBR, it would be impossible and very undesirable to use standardized test collection. In addition, the problems and biasses inherent in these collections often make the comparisons of systems somewhat questionable.

What to focus the evaluation on?

The evaluation of a Textual CBR system will largely be determined by the character and task of the system. The main difficulty in designing the evaluation and choosing the focus, though, is the fact that Textual CBR will often be an integral part of a larger system. The user’s perception will include the interface (like in SCALIR), or the performance of the textual CBR aspect will be a minor contribution to the overall system. In Intelligent Summarization, a similar problem occurs, and two different ways of assessing system performance, namely extrinsic and intrinsic evaluation, have been identified. Intrinsic evaluation measures how well the system works compared to an ”ideal system”, while extrinsic evaluation measures how well the overall system performs at a particular task (Jing *et al.* 1998). In particular in an industrial setting, it is impossible to describe

the performance of an "ideal system", so that it may be necessary to judge performance based on how useful the system is in when it being applied. Although the extrinsic evaluation of a system and the acceptance by the users are proof that the program works, but it does not identify the contribution of the Textual CBR approach chosen, and also does not allow researchers to compare it to their own work. Therefore, we think, that an intrinsic evaluation of the approach is still necessary.

To provide a better sense for the particular problems of evaluation Textual CBR systems, we will point out the evaluation strategies in a number of related projects, and discuss their strengths and weaknesses.

Evaluation of Textual CBR Systems

Similarly to our research, SPIRE (Daniels & Rissland 1997) tackles the case-representation bottleneck by identifying the best candidate passages for indexing-relevant information. The main focus of the system is to save the user time, and quickly bring his attention to the relevant passages of very long and complicated documents. Thus, it is appropriate to capture how successful this system is by measuring the average search length, which is defined as the average number of passages presented to the user before a relevant one (not necessarily all or the best one) is found. The standard of comparison involves manually generated, human queries. The standard of comparison is appropriate, but could have been improved, since the system's performance is compared to non-experts in the domain. Informal comparisons revealed no major advantages for either method. In order to give a more reliable analysis, statistical tests should have been carried out on the experimental results.

For purely retrieval-oriented systems, like FAQ-Finder (Burke *et al.* 1997), the task is to identify those cases relevant and sufficient for a user's information need. Here, it seems, precision and recall are the appropriate evaluation measures. There is, however, one major difference from the typical IR retrieval situation. In IR it is a success if a highly-ranked document is relevant to the query, since it is assumed that the user is interested in all relevant documents. The ranking is assumed to play a minor role. In help-desk or question-answering applications, however, it is important that one, if possible the most relevant document, is found - all other retrieved objects are not very important. Thus, a modified, and slightly relaxed variant of precision/recall is applied: a successful query is one where a relevant and sufficient document was ranked among the displayed, best documents. This definition of precision/recall makes it more difficult to judge the performance of FAQ-Finder as compared to traditional IR-systems. In addition, an ablation study was conducted and demonstrated that the novel semantic components of the system are useful.

An application closely related to Textual CBR is the indexing and subsequent use of text documents is the ASK-systems (Wisdo 1998). Here, evaluation would be more difficult. Stories to be used in an intelligent teaching environment are indexed by the user's question they are intended to answer. The CBR component is not separable from the overall system (since the retrieval of stories is tied to the user's questions in the course of acquiring knowledge, and the index assignment has to be assessed in the context of a model of the student's knowledge). The efficacy of the overall system can be determined by how well the students learn (for the evaluation of tutoring systems, see (Aleven 1997), and is some indication of the quality of the textual CBR in those systems, however, the evidence is only indirect and inconclusive. It is not possible reliably to assess the usefulness of the approach for other purposes or projects.

The FallQ project treats help-desk reports as cases, and finds the most similar cases related to a user's problem. Rather than using IR algorithms to compute similarity, similarity between cases is calculated as a function of shared words. In addition, semantically related words increase the similarity score between documents. Evaluation proves very difficult, because in the industrial setting, it is impossible to collect a set of queries examples with relevance judgements for the entire case-base. To overcome this problem, previous queries are rephrased and used as test cases, where the results of the original query is used as the standard of comparison. Although this is a way to create data for an evaluation, which would not be possible otherwise, it very much relies on the rephrasing of the queries, and a personal bias is hard to avoid. Also, the relevance judgements pertain only to the cases retrieved by the original query. In a preliminary set of experiments, the influence of semantic knowledge and natural language processing heuristics could be shown in an informal comparison of FallQ's retrieval with traditional IR methods. Overall, FallQ shows that the circumstances can make the formative evaluation of a Textual CBR system very difficult.

Another example of a system integrating CBR and IR techniques is SCALIR (Rose, 1994). It retrieves legal cases by spreading activation through a network of various, domain-specific nodes, connected by adaptive links. Apart from the hybrid reasoning and retrieval part, the system also has a novel interface, which was evaluated independently from the case-retrieval functionality. The relatively small-scale evaluation included 10 queries, where performance was measured by precision and recall. SCALIR's performance was compared against Westlaw, and a t-test could not find a statistically significant difference in performance between the two systems. The evaluation of SCALIR considered all three aspects for a Textual CBR system, in particular, it captured aspects of extrinsic and intrinsic system performance and used a good standard of comparison.

Our approach

In our Factor Finding project (Brüninghaus & Ashley 1997), we have been focussing on automatically deriving a representation for the cases from the texts, using text-classification methods. Since our data-set is relatively small, we use cross-validation to have more reliable data. As discussed above, we use accuracy as the evaluation measure, since in our domain, the non-assignment of categories is relevant. We compare the performance of the different algorithms to a default strategy, where the system always predicts that the factor does not apply. However, we also consider precision and recall, since they serve as an indicator whether the system does more than merely applying the default strategy. Our experiments were designed specifically to find out how well this transformation works; how often the system's assignments are correct is certainly the appropriate measure for this task.

Conclusion

While the evaluation is an important part of every research project, it is still an open issue how to carry it out for a textual CBR system. We discussed that choosing an evaluation measure, defining a standard of comparison and deciding what aspect of a system's performance to focus on are crucial decisions. We also laid out alternatives and influence factors to take into account for each of them.

References

Aleven, V. 1997. *Teaching Case-Based Argumentation Through a Model and Examples*. Ph.D. Dissertation, University of Pittsburgh, Pittsburgh, PA.

Brüninghaus, S., and Ashley, K. 1997. Using Machine Learning for Assigning Indices to Textual Cases. In *Proceedings of the 2nd International Conference on Case-Based Reasoning (ICCBR-97)*, 303–314.

Brüninghaus, S., and Ashley, K. 1998. Developing Mapping and Evaluation Techniques for Textual Case-Based Reasoning. In *Workshop Notes of the AAAI-98 Workshop on Textual CBR.*

Burke, R.; Hammond, K.; Kulyukin, V.; Lytinen, S.; Tomuro, N.; and Schoenberg, S. 1997. Question-answering from frequently-asked question files: Experiences with the faq finder system. Technical Report TR-97-05, Computer Science Department, University of Chicago, Chicago, IL.

1998.
CBR/IR-Discussion Group Prosition Statements. URL: `http://www.informatik.hu-berlin.de/~lenz-/discussion.html`.

Daniels, J., and Rissland, E. 1997. What you saw is what you want: Using cases to seed information retrieval. In *Proceedings of the 2nd International Conference on Case-Based Reasoning (ICCBR-97)*, 325–336.

Jing, H.; Barzilay, R.; McKeown, K.; and Elhadad, M. 1998. Summarizing Evaluation Methods: Experiments and Analysis. In *Working Notes of the AAAI-98 Spring Symposium on Intelligent Text Summzarization*, 60–68.

Koller, D., and Sahami, M. 1997. Hierarchically classifying documents with very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 170 – 178.

Lewis, D. 1995. Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 246–254.

Wisdo, C. 1998. A Scalable Approach to Question-Based Indexing of Encyclopedic Texts. In *Proceedings of the 2nd International Conference on Case-Based Reasoning (ICCBR-97)*, 200–210.

Yang, Y. 1997. An evaluation of statistical approaches to text categorization. Technical Report CMU-CS-97-127, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.