Jane Cleland-Huang

In the short amount of time available I make four points about the use and misuse of recall and precision as metrics.

1. **Context Matters:** The argument as to whether we should favor one metric (e.g., recall) over another (e.g., precision), or vice versa is really a moot point – because we need to understand the context of an algorithm's or tool's use in order to determine how it should be evaluated. In fact, the same algorithm, applied under different circumstances might need to be evaluated in different ways. Imagine for a minute an algorithm that dynamically constructs trace links between requirements and regulatory codes. If the algorithm is used to identify specific requirements that match a regulation and then to recommend those requirements to a human analyst – it would make sense to favor recall. The algorithm should find all matching requirements and allow the analyst to evaluate the recommendations, accepting the correct ones and rejecting the incorrect ones. On the other hand, imagine a scenario in which a human analyst has manually created links between requirements and regulations and wants to run the same algorithm to identify missing links. In this case, the analyst will not vet the generated links. The algorithm will be applied in a fully automated fashion and should only create links for which it has very high confidence. The first scenario favors the use of recall and the second precision – even though the algorithm being evaluated is the same. Only the context has changed.

2. **Neither Recall nor Precision is perfect:** The problem with both recall and precision as metrics is that they are computed at a fixed (sometimes rather arbitrary) threshold value. All artifacts above the threshold are considered recalled and there is no notion of ranking within this set. So neither recall nor precision are capable of differentiating between an algorithm that places all the targeted artifacts (e.g., trace links, requirements belonging to some category) at the very top of a ranked list or further down the list somewhere just above the threshold. Recall and precision are therefore poor predictors of performance. An example of a better metric is Mean Average Precision in which the precision of all targeted artifacts is computed and then averaged. This metric returns a score of 1 if all N targeted artifacts are at the top N positions in a list. Other metrics such as LAG (proposed by Jane Hayes) do similar things.

3. **Metric Abuse:** Our community is somewhat influenced in its use of metrics by other communities such as recommendation systems; however, the context is very different. Some of the lessons we have "learned" from those communities actually don't carry over very well. Consider the case of a recommender system that is used in Amazon. In this scenario – a very small improvement in the recommendation algorithm can lead to many thousands of dollars in increased sales. However, similar improvements in metrics, when applied to Requirements Engineering problems, don't necessarily lead to noticeable improvements. For example, it probably doesn't make much difference to a human analyst trying to identify non-functional requirements (aka quality requirements) in a requirements specification, if recall and precision improve by 1%. Their experience in performing the task has not significantly changed. The lesson here is that we need to really understand whether an improvement in an algorithm impacts the end use scenario. Alternately, we could imagine that if every researcher reports a

1% improvement – we will gradually creep forward and eventually, working together, will achieve a 5% or 10% improvement. Unfortunately it often doesn't work that way because what we find is that one algorithm (A) outperforms another algorithm (B) – but only on certain datasets, and papers tend to report improvements associated with specific datasets. We therefore need metrics and benchmarks that measure progress across multiple datasets and be careful how we use metrics to report results.

4. **Impact upon the end user:** This point really reiterates my previous one. Metrics are great for quantifying results; however, we need to keep the end user in mind and evaluate our proposed solutions with respect to their impact upon actual usage tasks.

As I attended the panel remotely, I prerecorded a short statement (in case of technical difficulties). That statement can be found here: http://sarec.nd.edu/REPanel.mp4