Evaluation of example tools for hairy tasks.

Presenter: Hardik Sahi (20743327)

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS

Outline

- Definition of a hairy task.
- Metrics for tool evaluation
- When is recall favoured over precision?
- New metrics: Weighted F measure, Summarization
- Purpose of project
- Case study 1: Re-evaluation of Paper 1
- Case study 2: Re-evaluation of Paper 2
- Conclusion
- References

What is a hairy RE or SE task?

A hairy task is defined as follows:

- A task that can be done manually on a small scale but becomes unmanageable on large scale. e.g. Analyzing app reviews, Finding ambiguities in RE documents.
- For such tasks, humans need tool assistance.
- The tool should be such that it does not miss any true positives (equivalently, has minimum false negatives).

Metrics for tool evaluation [1]

Precision: What proportion of positive identifications by the tool are actually correct?

Recall: What proportion of actual positives were identified correctly?

F1 measure: Harmonic mean of Precision and Recall.

	Relevant	Not Relevant
Found	True Positive (TP)	False Positive (FP)
Not Found	False Negative (FN)	True Negative (TN)

Precision (P) = TP/ (TP+FP)

Recall (R) = TP/(TP+FN)

F1 measure = 2*P*R/(P+R)

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS

When is recall favoured over precision?

Consider a tool which is supposed to assist humans in tackling a High Dependency (HD) task:

Cost of missing a TP => Manually go through all the documents. (Very expensive)

Cost of rejecting a FP => Manually go through only a small subset of results returned by tool (Not expensive)

This calls for evaluating tools using metrics that **favour recall more than precision.**

New metrics to evaluate tools [2]

- **Weighted F measure**:
  (F1 measure is P and R weighted equally)

$$F_\beta = (1+\beta^2)*\frac{P*R}{(\beta^2*P) + R}$$

where β is the ratio of importance of R and P.

- **Summarization**:
  Fraction of original doc eliminated by the tool.

  Human can perform exact same task on a much smaller output of tool.

$$S=\frac{TN+FN}{TN+FN+TP+FP} = \frac{Not\ returned\ by\ tool}{Total\ possibilities}$$

A tool is really good at performing hairy task if:

- Has high recall
- Has high summarization.

Determining ß

$$\beta_T = \frac{\text{Average time to manually } \underline{\textbf{find}} \text{ correct answer among potential answers } (\zeta_{find})}{\text{Average time to manually } \underline{\textbf{determine}} \text{ whether potential answer is corect } (\zeta_{det})}$$

$$= \frac{1}{\text{Fraction of correct answers among potential answers } (\lambda)}$$

$$\beta_{T,t} = \frac{\text{Average time to manually } \underline{\textbf{find}} \text{ correct answer among potential answers } (\zeta_{find})}{\text{Average time to manually } \underline{\textbf{vet}} \text{ a potential answer tool } t \text{ returns } (\zeta_{vet})}$$

The above values of ß are calculated empirically. They are then used to calculate weighted F measure.

Purpose of the project

- Analyze papers that detail working and evaluation of natural language based tools for hairy tasks.
- Check whether the proposed evaluation metrics make sense.
- If not, re-evaluate the tools using empirical evidence presented in the paper.

Paper 1: Using Tools to Assist Identification of Non requirements in Requirements Specifications – A Controlled Experiment [3]

- Proposes a Neural Network based tool that labels text fragments as **requirements or non-requirements (Information)**.
- Issues warnings when predicted label does not match the actual label **(Defect)**.
- Controlled study where 2 groups of students identify defects in 2 requirements documents with and without tool.

Paper 1 : Understanding confusion matrix

	Actual	Predicted	Impact
True positive (TP)	Defect	Defect	Correct warning
True negative (TN)	No defect	No defect	No warning
False positive (FP)	No defect	Defect	**False warning**
False negative (FN)	Defect	No defect	**Missed warning**

Cost of handling FN is prohibitive as Requirements Engineer has to manually go through entire document to identify any missed defect.

If the tool issues way too many FP, the engineers waste a lot of their time rejecting them.

Paper 1 : What authors say?

“ The results indicate that given high accuracy of the provided warnings, users of our tool are able to perform slightly better than the users performing manual review. They managed to find more defects, introduce less new defects, and did so in shorter time. However, when many false warnings are issued, the situation may be reversed. Thus, the actual benefit is largely dependent on the performance of the underlying classifier. False negatives (i.e., defects with no warnings) are an issue as well, since users tend to focus less on elements with no warnings ” [3]

Paper 1 : My analysis

	Wiper control	Window lift
Total elements	115	261
Total requirements	85	186
Total information	30	75
Total defects	20	17
Total warnings	24	70
Correct warnings	12	12
Unwarned defects	8	5
Accuracy	82.6%	75.8%

$$\lambda_{WC}=\frac{20}{115}=0.174$$

$$\beta_T=\frac{1}{\lambda_{WC}}=\frac{115}{20}=5.75$$

$$P_{WC}=\frac{12}{24}=0.5$$

$$R_{WC}=\frac{12}{20}=0.6$$

$$F_{5.75}=0.596 \approx R_{WC}$$

$$\text{From the paper, } \zeta_{find}=16.6\text{s/element}$$

$$\text{So, } \zeta_{det}=\frac{16.6}{5.75}=2.88\text{s/element}$$

$$\lambda_{WL}=\frac{17}{261}=0.0651$$

$$\beta_T=\frac{1}{\lambda_{WL}}=\frac{261}{17}=15.352$$

$$P_{WL}=\frac{12}{70}=0.1714$$

$$R_{WL}=\frac{12}{17}=0.705$$

$$F_{15.352}=0.6958 \approx R_{WL}$$

$$\text{From the paper, } \zeta_{find}=9\text{s/element}$$

$$\text{So, } \zeta_{det}=\frac{9}{15.352}=0.586\text{s/element}$$

Paper 1 : My conclusion

- The values of ß (>>1) indicate that authors should pay more attention to recall over precision.
- This is further cemented by the fact that cost associated with manually telling whether answer is correct is significantly smaller than manually finding out correct answers out of all potential answers.

So,

The idea of the authors that the usability of tool is heavily dependant on tool not giving way too many false warnings (FP) and not missing actual defects (FN) is correct and supported by above calculations. BUT..

Authors should focus on recall and not accuracy to ensure that their tool is useful.

Paper 2 : Finding and Analyzing App Reviews Related to Specific Features: A Research Preview [4]

- Proposes a ML based tool that:
  - **Input**: Line describing a feature.
  - **Output**:
    - Find reviews that refer to a specific feature.
    - Identify bug reports, change requests and users' sentiment about this feature
    - Visualize and compare feedback for different features in a dashboard

Paper 2 : Understanding confusion matrix

	Actual	Predicted	Impact
True positive (TP)	Review related to feature	Review returned	Correct action taken
True negative (TN)	Review NOT related to feature	Review not returned	Correct action taken
False positive (FP)	Review NOT related to feature	Review returned	**False review returned**
False negative (FN)	Review related to feature	Review not returned	**Missed Review**

Paper 2: What authors say?

“ We evaluated our prototype using 10-fold cross-validation and obtained precision of 0.360, recall of 0.257 and F1 score of 0.300. We observed that for queries formed by two keywords (e.g. add reservation) and term proximity less of than three words, the approach achieve precision at the level of 0.88. ”

Paper 2: My analysis

The paper does not provide any data to conduct analysis.

The authors should collect the following data to enable empirical analysis :

- Frequency of related (correct) reviews out of total 200 reviews (Lambda)
- Time taken to go through all the reviews manually (Numerator of beta)
- How was ground truth created? How many people were involved in it?

Once we have access to the above information, we can perform detailed empirical analysis and quantitatively derive meaningful results.

Paper 2: My conclusion

The task of extracting app reviews relevant to a feature is a hairy one as it is very expensive when done on a large scale (100 vs 10000 reviews).

Cost of correcting False Negatives (FN) is prohibitive as this would mean analyzing all the reviews manually, effectively rendering the tool useless.

So,

Authors evaluate their tool using F1 measure (equal emphasis to P and R) probably out of habit (inspired from IR) OR by not understanding the above mentioned points.

This is a wrong metric for evaluation and should be replaced with weighted F measure.

Conclusion

- Most of the SE / RE tasks involving natural language are hairy.
- Sometimes, authors use conventional F1 or precision metrics to evaluate their tools without considering that that very usefulness of their tool is heavily dependant on high recall.
- Each task must to thoroughly analyzed to decide which metric to use - Recall, Weighted F measure, Summarization etc.

References

1. https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall
2. https://cs.uwaterloo.ca/~dberry/FTP_SITE/tech.reports/EvalPaper.pdf
3. https://link.springer.com/chapter/10.1007/978-3-319-77243-1_4
4. https://link.springer.com/chapter/10.1007/978-3-030-15538-4_14

Thank You

Any Questions?