RE for AI: What is an RS for an AI?

Daniel M. Berry
University of Waterloo

Tasks Requiring Intelligence

We are talking about *tasks* requiring

***real intelligence (RI)*, i.e.,**

from a *human*.

The task is to find *correct answers*

in a space of *answers*,

some *correct* and the rest *incorrect*.

Building an AI or LM

We want to build an *artificial intelligence (AI)* that does the task.

This AI might be a *learned machine (LM)* which is the result of *machine learning (ML)*, whether it is taught, self-teaching, or both.

I use "AI" to mean *any* of them.

Specifying Requirements of AI

How do we *specify the requirements* of the AI in a way that …

when we have an *implementaion* of the AI,

we can use the

***requirements specification (RS)* of the AI**

to decide whether the

implementation *statisfies*

the AI's *requirements*?

How are AIs Described?

Notice, I did not say "How are AIs *specificd*?"

No AI worker expects to describe an AI's behavior completely.

Everyone uses instead, imprecise terms,

e.g., "usually", "probably", "approximately", etc.,

They give only empirically determined probabilities.

Evaluating AIs

An AI is evaluated with *vague* measures,

such as recall and precision.

Vagueness

A measure is *vague* iff

There is little certainty on what values of the measure are good and bad, and …

even when it is certain that some value is good and another value is bad,

it is not certain what value in between is the *boundary* between the good and bad.

Useless Specs

While there might be a simple specification of a task, e.g.,

“Return only images that contain stop signs.”,

there is no actionable specification that

identifies all and only images containing stop signs.

No Formal Specs

Thus, there is no possibility of a formal mathematical specification.

And yet, we want to be able to say with certainty whether …

an implementation of an AI for a task does indeed do the task at least well enough.

How to do that seems not to be satisfactorily answered in the literature

Research Question

How does one write a requirements specification (RS), S**,**

for an AI, A**, for a task,** T**,**

in a way that S **can be used to decide whether**

A **correctly implements** T**,**

by asking whether

A **satisfies** S**?**

Special Case of an LM

If A is an LM — a data-centric system —

S includes the real-world (RW) data that taught A all it knows.

Key Observation

Fundamentally,

an AI for a task

must *mimic*

humans

who are using their RI to do the same task

[Acknowledge: Alessio Ferrari].

Human Behavior Is Correctness

When we don't have a complete specification of the task,

we accept that

what humans do in practice,

as measured empirically,

is correct.

So, we're heading to an empirical determination of correctness.

Imperfect Mimicry

This mimicry will almost never be perfect.

Thus, an RS for an AI doing the task must

describe this mimicry in a way that

allows *measuring how well* the AI mimics humans.

[Acknowledge: Vogelsang and Borg]

Vague Measures

These measures are vague

i.e., whether their values are satisfactory

will not have “yes” and “no” answers.

A decision of how well

the AI mimics humans

will be a matter of judgment.

Some Sets of Measures

Two such sets of measures are

1. ***recall and precision*, measures of frequency of correctness**

   w.r.t. a *human-determined* gold set.

2. **interrater agreement on the task,**

   comparing the AI and some humans.

There are other sets of measures that achieve the same objective.

Empirical, Not Logical

The measures are not binary, and …

human performance is part of the decision.

Therefore, the truth of

“evaluation criteria are met”

≡

“RS is satisfied”

is not logical, but is empirical.

As With Zave & Jackson

Just as with the Zave–Jackson Validation Formula (ZJVF),

which is about systems that interact with the RW.

$D, S \vdash R$

As With ZJVF

$D, S \vdash R$

“⊢” is “entailment”

D **and** R**, about RW are informal**

S **about PL program is formal, but**

S **about molectular SW or an AI is informal, being about RW.**

Can Skip For Many Audiences

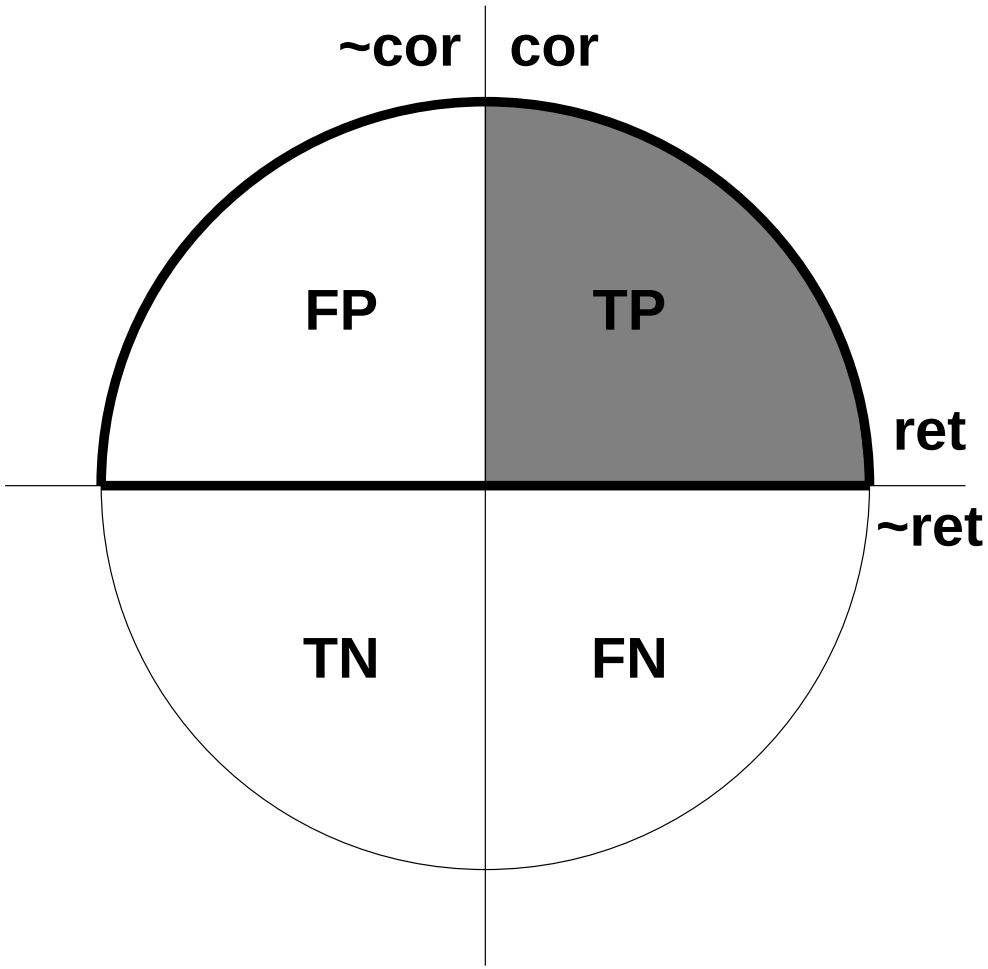
Precision

***P* is the percentage of the tool-returned answers that are correct.**

$$P = \frac{|\ ret \cap cor\ |}{|\ ret\ |}$$

$$= \frac{|\ TP\ |}{|\ FP\ | + |\ TP\ |}$$

Precision



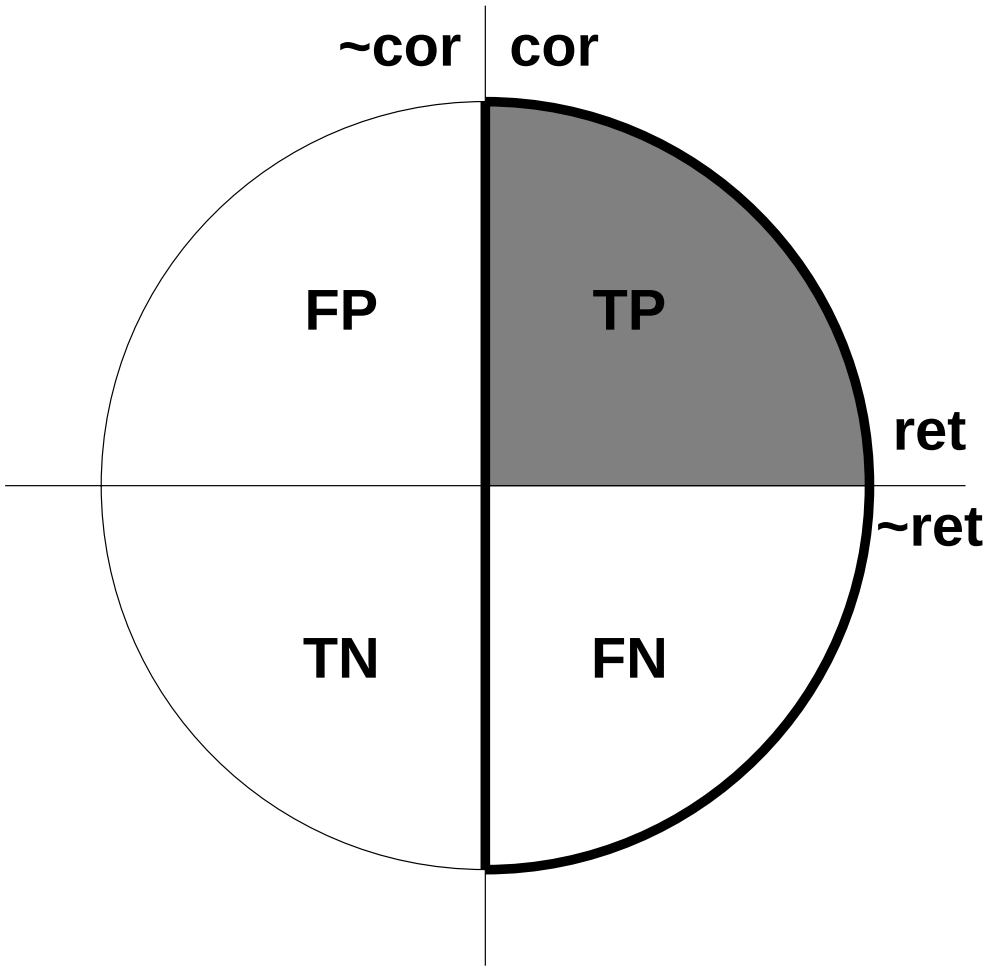
Recall

***R* is the percentage of the correct answers that the tool returns.**

$$R = \frac{|\ ret \cap cor\ |}{|\ cor\ |}$$

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

Recall



F-Measure

F*-measure: harmonic mean of *P* and *R
(harmonic mean is the reciprocal of the arithmetic mean of the reciprocals)

Popularly used as a composite measure.

$$F = \frac{1}{\frac{\frac{1}{P} + \frac{1}{R}}{2}} = 2 \cdot \frac{P \cdot R}{P + R}$$

Weighted *F*-Measure

For situations in which *R* and *P* are not equally important, there is a weighted version of the *F*-measure:

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

Here, β is the ratio by which it is desired to weight *R* more than *P*.

Note That

$F = F_1$

As β grows, F_β approaches R (and P becomes irrelevant).

If Recall Very Very Important

Now, as $\beta \to \infty$,

$$F_\beta \approx \beta^2 \cdot \frac{P \cdot R}{\beta^2 \cdot P}$$

$$= \frac{\beta^2 \cdot P \cdot R}{\beta^2 \cdot P} = R$$

As the weight of *R* goes up, the F-measure begins to approximate simply *R* !

If Precision Very Very Important

Then, as $\beta \to 0$,

$$F_\beta \approx 1 \cdot \frac{P \cdot R}{R}$$

$$= P$$

which is what we expect.

R vs *P* Tradeoff

_P_ and _R_ can usually be traded off in an IR algorithm:

- **increase _R_ at the cost of decreasing _P_, or**
- **increase _P_ at the cost of decreasing _R_**

Extremes of Tradeoff

Extremes of this tradeoff are:

1. **tool returns all possible answers, correct and incorrect: for**

   $R = 100\%$, $P = C$,

   where $C = \frac{\#\ \text{correctAnswers}}{\#\ \text{answers}}$

2. **tool returns only one answer, a correct one: for**

   $P = 100\%$, $R = \varepsilon$,

   where $\varepsilon = \frac{1}{\#\ \text{correctAnswers}}$

Extremes are Useless

Extremes are useless, because in either case, …

the entire task must be done manually on the original document in order to find *exactly* the correct answers.

100% Recall Useless?

Returning everything to get 100% *R* doesn't save any real work, because we still have to manually search the entire document.

This is why we are wary of claims of 100% *R* … Maybe it's a case of this phenomenon!

What is missing?

Summarization

Different Summarization

The summarization I define is different from any semantics-based summarization that you may be thinking of.

I am telling you this now so that you are not surprised when I don’t use it in the way you expected.

Summarization

If we can return a subdocument significantly smaller than the original …

that contains *all* correct answers, …

then we have saved some real work.

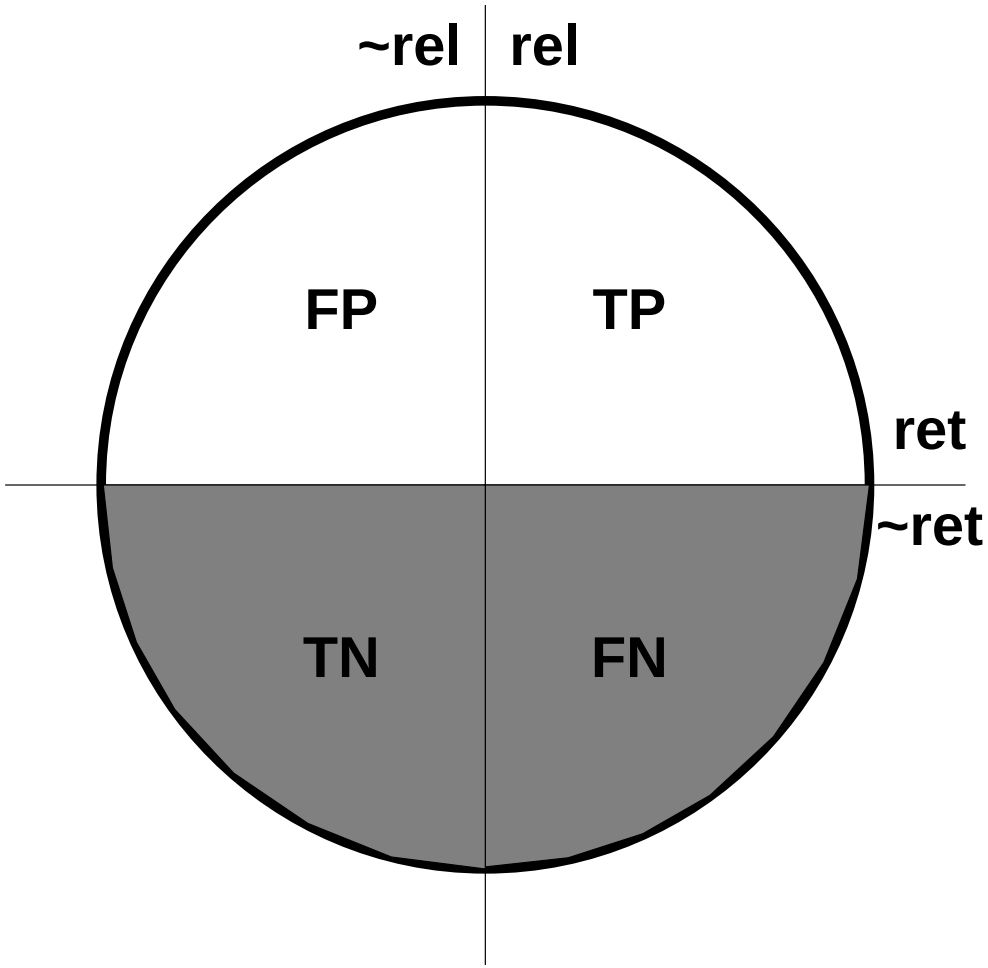
The *remaining* manual task will take significantly less time than the original, pre-tool-application manual task.

Summarization Measure

Summarization = fraction of the original document that is eliminated in what is returned

$$S = \frac{|\sim ret|}{|\sim ret \cup ret|} = \frac{|\sim ret|}{|\sim rel \cup rel|}$$

$$= \frac{|TN| + |FN|}{|TN| + |FN| + |TP| + |FP|}$$


~rel
rel
FP
TP
ret
~ret
TN
FN

How to Use Summarization

If there is no escaping doing the task, and the alternative to using a tool is to do the task manually, then …

we would *love* a tool with 100% *R* and 90% *S*.

Then we really do not care about *P*,

With high *S*, the time to vet the tool's output will be significantly smaller than the time to do the task entirely manually, *and* …

we end up with potentially the same 100% *R*.

In Other Words

That is, if we can get rid of 90% of the document with the assurance that …
what is gotten rid of contains *only irrelevant* items and thus …

what is returned contains *all* the relevant items, and

the time to vet the output is only 10% of the time to do the task manually on the input document,

then we are *very happy*! ☺

End of Skip

Remember:

For an AI, *A*

Recall (*R*): percentage of the correct answers that are returned by *A*

Precision (*P*): percentage of the answers returned by *A* that are correct

Summarization (*S*): percentage of the input to *A* that are removed in the output that *A* returns, i.e., (100% - (size(output) / size(input)))

Remember, Cont'd

Correctness is *all* and *only* the correct answers.

So, *R* and *P* are the two sides of "*all* and *only*":

***R* measures how close to *all* the correct answers are in the output.**

***P* measures how close to *only* the correct answers are in the output.**

First Stop-Sign Finder AI

One AI, *A*1, to classify each image,

Has_a_Stop_Sign or not,

and to output only those that have at least one stop sign.

First Stop-Sign Finder AI

***A*1 picks out of thousands of images the few that contain stop signs in order**

to produce a training set of images,

each of which is correctly classified as to whether it contains a stop sign,

to be used to train another AI, *A*2,

to be used in an autonomous vehicle (AV) to identify stop signs in real time.

Apply R, P, and S to $A1$

1000 images of which 200 have stop signs

***A*1 returns 400 images of which 190 truly have stop signs**

***R* =**

Apply R, P, and S to $A1$

1000 images of which 200 have stop signs

***A*1 returns 400 images of which 190 truly have stop signs**

***R* = 190/200 = 95%**

***P* =**

Apply R, P, and S to $A1$

1000 images of which 200 have stop signs

***A*1 returns 400 images of which 190 truly have stop signs**

***R* = 190/200 = 95%**

***P* = 190/400 = 47.5%**

***S* =**

Apply R, P, and S to $A1$

1000 images of which 200 have stop signs

***A*1 returns 400 images of which 190 truly have stop signs**

***R* = 190/200 = 95%**

***P* = 190/400 = 47.5%**

***S* = 600/1000 = 60%**

Not Bad If …

***R* = 190/200 = 95%**

***P* = 190/400 = 47.5%**

***S* = 600/1000 = 60%**

Not a bad situation to be in, if …

Not Bad If …

***R* = 190/200 = 95%**

***P* = 190/400 = 47.5%**

***S* = 600/1000 = 60%**

Not a bad situation to be in, if …

Humans have poorer than 95% recall!

Second Stop-Sign Finder AI

Another AI, *A2*, to classify each of a continuous stream of images,

Has_a_Stop_Sign or not,

and if *A2* sees a stop sign,

it signals to the AV to stop at the stop sign.

Evaluation of Each AI

Each AI is evaluated by its *R* and *P*,

w.r.t. a manually developed gold set of classified images.

Humanly-Achievable *R* and *P*

Each human participating in developing the gold set computes his or her own *R* and *P*,

and the *averages* of their *R* and *P* values are

the humanly-achievable-recall (HAR) and

the humanly-achievable-precision (HAP)

of the stop-sign recognition task.

HAR and HAP

Andreas Vogelsang observes that there may be contexts in which

minima,
maxima,
modes, or
medians

should be used instead of averages.

HAR and HAP

Each of the HAR and HAP of the stop-sign recognition task is probably about 99.99%.

But could probably get very accurate data from `www.captcha.net`, ☺.

Basic *A*1 Requirement

For *A*1, the basic requirement is to achieve *R* of 100% and *P* of 100%,

because any *R* and *P* less than 100% means

Basic *A*1 Requirement

For *A*1, the basic requirement is to achieve *R* of 100% and *P* of 100%,

because any *R* and *P* less than 100% means

that the data on which *A*2 trains are flawed,

and *A*2 will not learn perfectly,

setting up possible AV failures.

Basic *A*2 Requirement

For *A*2, the basic requirement is to achieve *R* of 100% and *P* of 100%,

because any *R* less than 100%, e.g., *R′* means

Basic *A*2 Requirement

For *A*2, the basic requirement is to achieve *R* of 100% and *P* of 100%,

because any *R* less than 100%, e.g., *R′* means

that the AV will fail to stop at 100% – *R′* of the stop signs.

Basic *A*2 Requirement

For *A*2, the basic requirement is to achieve *R* of 100% and *P* of 100%,

because any *P* less than 100%, e.g., *P′* means

that the AV will stop unnecessarily 100% – *P′* of the time.

Good or bad?

Basic *A*2 Requirement

For *A*2, the basic requirement is to achieve *R* of 100% and *P* of 100%,

because any *P* less than 100%, e.g., *P′* means

that the AV will stop unnecessarily 100% – *P′* of the time.

Good or bad?

Might lead to more rear-end collisions!

100% R and P Achievable?

But, are *R* = 100% and *P* = 100% achievable?

100% R and P Achievable?

But, are *R* = 100% and *P* = 100% achievable?

No

Are *R* = 100% and *P* = 100% even reasonable to expect?

100% R and P Achievable?

But, are *R* = 100% and *P* = 100% achievable?

No

Are *R* = 100% and *P* = 100% even reasonable to expect?

No

What *Is* Achievable?

So what *are* reasonable *R* and *P* to expect to achieve and maybe beat?

What *Is* Achievable?

So what *are* reasonable *R* and *P* to expect to achieve and maybe beat?

HAR and HAP (what is humanly achievable)

What *Is* Achievable?

So what *are* reasonable *R* and *P* to expect to achieve and maybe beat?

HAR and HAP (what is humanly achievable)

For either AI, the basic requirement is

- **for its *R* to achieve or beat the task's HAR and**
- **for its *P* to achieve or beat the task's HAP.**

Why HAP and HAP Acceptable?

Why are we satisfied with achieving or beating HAR and HAP?

Accidents are inevitable, particularly if humans are doing the task.

If an AI's *R* and *P* achieve or beat the task's HAR and HAP, then it will have no more accidents than a human doing the task.

Why HAP and HAP Acceptable?

While no accident is good, society can accept an AI's doing a task if the AI will have no more or fewer accidents than a human will.

Vetting of an AI's Output?

The output of *A*1 is vetted (checked) by human beings.

Vetting of an AI's Output?

The output of *A*1 is vetted (checked) by human beings.

There is plenty of time, and vetting improves overall *R* and *P*.

The output of *A*2 is *not* vetted by human beings.

Vetting of an AI's Output?

The output of *A*1 is vetted (checked) by human beings.

There is plenty of time, and vetting improves overall *R* and *P*.

The output of *A*2 is *not* vetted by human beings.

There is no time, because the output is needed *immediately* by its AV.

Vetting for *A*2 Is Nonsense

Putting a human in an AV to vet *A*2

- **takes away the AV's A (autonomy) and**
- **slows the AV's response time to the point of uselessness.**

Amended Basic Requirement

For *each* of *A*1 and *A*2,

> ***R* should be at least as good as the task's HAR, and**
>
> ***P* should be at least as good as the task's HAP.**

Can we do better with *both R* and *P*?

Amended Basic Requirement

For *each* of *A*1 and *A*2,

> ***R* should be at least as good as the task's HAR, and**
>
> ***P* should be at least as good as the task's HAP.**

Can we do better with *both R* and *P*?

Unlikely! Why?

Tradeoff

It's unlikely because of the usually inevitable tradeoff:

In implementing any AI, a higher *R* can be achieved at the cost of lowering *P*, and vice versa:

Tradeoff

A less strict classification causes more images to be accepted as containing a stop sign, thus

increasing *R*, and unfortunately,

increasing imprecision (= 100% – *P*).

Tradeoff

A more strict classification causes fewer images to be accepted as containing a stop sign, thus

increasing *P*, and unfortunately,

decreasing *R*.

Usually Recall is Critical

Regardless of how critical a high *P* is,

a high *R* is very critical.

Finding *all* correct answers is often *very* necessary.

Lives depend on doing so.

If finding *all* correct answers were not necessary, we would not bother building an AI for doing it.

If High *R* is Important

However, if achieving high *R* is important, there may be *no* choice but to accept low *P*.

For *A*1, what does low *P* mean?

If High *R* is Important

However, if achieving high *R* is important, there may be *no* choice but to accept low *P*.

For *A*1, what does low *P* mean?

Lots of false positives among the output of *A*1

Are these false positives dangerous?

If High *R* is Important

However, if achieving high *R* is important, there may be *no* choice but to accept low *P*.

For *A*1, what does low *P* mean?

Lots of false positives among the output of *A*1

Are these false positives dangerous?

No

Effect of False +s on Vetters

What effect do lots of false positives among the output of *A*1 have on vetters?

Very discouraging

Effect of False +s on Vetters

How tolerable is low *P* for *A*1?

Effect of False +s on Vetters

How tolerable is low *P* for *A*1?

***if S* is high?**

Effect of False +s on Vetters

How tolerable is low *P* for *A*1?

***if S* is high?**

very tolerable

***if S* is low?**

Effect of False +s on Vetters

How tolerable is low *P* for *A*1?

***if S* is high?**

very tolerable

***if S* is low?**

not tolerable

Low P for $A1$

In the end, for *A*1,

if the effective *R* of *A*1 determined after vetting beats the task's HAR,

and the time to vet *A*1's output is less than the time to do the classification task manually,

in other words, to effectively vet *A*1's input,

then *A*1 meets its requirements.

Low *P* for *A*1

After all, since the task of *A*1 is essential,

the alternative to running *A*1 is

to do the task completely manually…

at the cost of a lot *more* tedious, *boring* grunt work!

yechhhhh!!!!

Effect of False +s on AVs

For *A*2, what does low *P* mean?

Remember that there is *no* vetting.
***A*2 runs in an AV.**

Effect of False +s on AVs

For *A2*, what does low *P* mean?

Remember that there is *no* vetting.
***A2* runs in an AV.**

Lots of unnecessary stops by the AV!

Are these unnecessary stops by the AV dangerous?

Effect of False +s on AVs

For *A2*, what does low *P* mean?

Remember that there is *no* vetting.
***A2* runs in an AV.**

Lots of unnecessary stops by the AV!

Are these unnecessary stops by the AV dangerous?

Could very well be; could lead to lots of rear-end-from-front-end collisions!

Low P for $A2$

How tolerable is low *P* for *A2*?

Low *P* for *A*2

How tolerable is low *P* for *A2*?

Definitely not!

Low *P* for *A*2

How tolerable is low *P* for *A2*?

Definitely not!

In the end, for *A2*,

if *R* of *A2* achieves or beats the HAR,

and *P* of *A2* achieves or beats the HAP,

then *A2* meets its requirements.

Ease of Implementation

So,

implementation of *A*1

will be significantly easier than

implementation of *A*2.

Conclusion

It’s clear that a requirements specification (RS) for an AI needs more than just *R* and *P*.

Needs More Than *R* And *P*

RS needs also:

- **HAR of the task, with which to compare *R*,**
- **HAP of the task, with which to compare *P*,**
- **β of the context of the AI, the ratio by which to weight *R* more than *P*, and**
- **in a case in which vetting is possible or required, *S* of the AI, to help evaluate the tradeoff between *R* and *P*.**

Requires Understanding Context

The second last value requires a *full* understanding of the context in which the AI is being used, including

- **the cost of a false negative, and**
- **the cost of a false positive**

in the context.

Decision Not Cut And Dry

Finally, the decision of whether

the AI satisfies its RS and meets its requirements

will involve

engineering judgement and

evaluation of tradeoffs in the AI's context,

and will *not* be a simple "yes" vs "no" decision.

Decision Not Cut And Dry

The decision is not cut and dry because of all of the vague elements in the RS.

The RS for an AI is as vague as are fitness criteria for vague qualitative, non-functional requirements, e.g., “fast response time” or “friendly user interface”.

Vagueness in the RS

For examples:

1. How much *S* is enough for vetters to *tolerate* having to vet in the presence of *low P*?

2. How *critical* must the task be in order that the *only* alternative to an AI that satisfies its RS is doing the task manually?

3. How much *S* is enough for vetters to *tolerate* having to vet *more than* having to do the task manually?

Vagueness in the RS

What is the interaction between the previous three questions?

What is to be done in the situation in which the task is *fairly critical*, the AI *just misses achieving* the task's HAR and HAP, but the *S* is *very high*?

Another Skip

Continually Learning LM

Consider evaluating *R* and *P* using a gold set for a LM, *M*, that learns from its mistakes.

What is the main problem with evaluating every new version of *M* with the same gold set?

Continually Learning LM

Consider evaluating *R* and *P* using a gold set for a LM, *M*, that learns from its mistakes.

What is the main problem with evaluating every new version of *M* with the same gold set?

***M* learns from its mistakes on the gold set, and the gold set loses validity as an instrument.**

Continually Learning LM

OK, So we use a different gold set for each new version of *M*!

What is wrong with doing so?

Continually Learning LM

OK, So we use a different gold set for each new version of *M*!

What is wrong with doing so?

We lose comparability of the evaluations that comes from knowing that all versions are evaluated against the same gold set,

and now have to trust that all gold sets are equally difficult to any LM.

Continually Learning LM

So what do we have to do to preserve validity *and* comparability?

Continually Learning LM

So what do we have to do to preserve validity *and* comparability?

Somehow inhibit *M*'s learning from its mistakes on the gold set.

If *R* & *P* Meaningless for Task?

If the task for an AI does not lend itself to being evaluated by *R* and *P*, then …

End Skip

Lessons from my Son's Startup

My son's start up is developing an AI that will make life-critical medical decisions from data,

…

decisions that are difficult for humans to make because of the large volume of data that are relevant.

We want both *R* and *P* to be 100%,
or at least beating HAR and HAP,

for patients' sakes.

Several High *R* AIs

They have several high-*P* AIs, *P* very close to HAP.

But none has *R* better than 50%. Yuck ☹ !

But

The region of recall of each pair of AIs overlaps only a little.

So try running them all to see if

the union of their outputs

has *R* that beats HAR!

Computers and running software are cheap!

If *R* & *P* Meaningless for Task?

If the task for an AI does not lend itself to being evaluated by *R* and *P*, then …

use the traditional measures for evaluating the task, but …

If *R* & *P* Meaningless for Task?

If the task for an AI does not lend itself to being evaluated by *R* and *P*, then …

use the traditional measures for evaluating the task, but …

also evaluate expert humans doing the task with these measures, and …

If *R* & *P* Meaningless for Task?

If the task for an AI does not lend itself to being evaluated by *R* and *P*, then …

use the traditional measures for evaluating the task, but …

also evaluate expert humans doing the task with these measures, and …

ascertain that the AI is at least as good as the humans.

More General Formulation

What a Spec of an AI is

It is clear that an RS for an AI needs more than the measures

M_1, …, and M_n

that are used to evaluate the AI.

The RS needs also …

RS Needs Also

- **humanly achievable values of M_1, …, and M_n for the task,**

  with which to compare the AI's M_1, …, and M_n values.

- **relative importance of the individual measures M_1, …, and M_n**

  to help evaluate tradeoffs between M_1, …, and M_n,

RS Needs Also, Cont'd

- **when vetting is possible or required, the *S* of the AI,**

  to help evaluate tradeoffs between M_1, …, and M_n, and

- **any data, e.g., training data, that the AI needs to function correctly.**

Context Dependencies

Calculating relative importance of the individual measures M_1, …, and M_n requires

***full* understanding of the context in which the AI is being used, including**

cost of achieving a high value in each of the individual measures M_1, …, and M_n,

in the context.

NFRs will help define the context and decide the tradeoffs.

Engineeering Judgement

Finally, the decision of whether

the AI satisfies its RS and meets its requirements

will involve engineering judgement and

evaluation of tradeoffs in the AI's context.

Ii will *not* be a simple "yes" vs "no" decision,

because of all of the vague elements in the RS.

RS Like Fitness Criteria

The RS for an AI is as vague as are

fitness criteria for vague NFRs e.g.,

“fast response time” or “friendly user interface”.

Examples

1. **How big must *S* be for vetters to *tolerate* having to vet when any of M_1, …, and M_n has a *low* value?**

2. **How *critical* must the task be in order that the *only* alternative to an AI that satisfies its RS is doing the task manually?**

3. **How much *S* is enough for vetters to *tolerate* having to vet because vetting will cost less than doing do the task manually?**

Engineering

These questions can interact in an engineering way.

For example, what to do done when

the task is *fairly critical*,

the AI *just misses achieving* the task's humanly achievable measures,

but

***S* is *very high*?**

Acknowledgments

Thanks to Jo Atlee for her comments on an earlier version of these slides.

Thanks to Alession Ferrari and Andreas Vogelsang for good discussions.