Apêndice B

Diretrizes de avaliação

Apêndice do Capítulo 14

Brielen Madureira

Publicado em: 13/03/2024

B.1 Diretrizes de avaliação para um modelo de cinco estágios

Critérios extraídos de (Cohen; Howe, 1988), traduzidos com substituição de "pesquisa" para "projeto", de forma a para torná-los mais genéricos, ou seja, se referir ao desenvolvimento de projetos em geral e não apenas aos de pesquisa em inteligência artificial.

Avaliando a tarefa ou objetivo do projeto

1. A tarefa é significante? Por quê?
    - Se o problema já foi definido antes, sua reformulação traz melhorias?
2. Há chances de seu projeto contribuir de forma significativa para o problema? A tarefa é manejável?
3. Conforme a tarefa se torna mais específica para seu projeto, ela ainda é representativa de uma classe de tarefas?
4. Algum aspecto interessante do problema foi simplificado ou abstraído?
    - Se o problema já estava previamente definido, algum aspecto dessa definição anterior foi abstraído ou simplificado?
5. Quais são os objetivos intermediários do projeto? Quais tarefas-chave serão ou já foram resolvidas como parte do projeto?
6. Como você saberá quando tiver demonstrado satisfatoriamente uma solução para o problema? É possível, para a tarefa em questão, demonstrar que se chegou a uma solução?

Critérios para avaliação de métodos

1. Que melhorias o método traz sobre tecnologias existentes?
    - Ele cobre mais situações (*input*)?
    - Ele produz mais variedade de comportamentos desejáveis (*outputs*)?

- Espera-se que ele seja mais eficiente em termos de memória, velocidade, tempo de desenvolvimento, etc.?
- Ele é promissor para futuros desenvolvimentos (por exemplo, se um novo paradigma for introduzido)?

2. Há uma métrica já estabelecida para avaliar a performance do método (por exemplo, ela é normativa, tem validade cognitiva)?
3. O método depende de outros métodos? Ou seja, há necessidade de pré-processamento de dados, acesso a certas bases de conhecimento ou rotinas?
4. Quais são as premissas ou suposições necessárias?
5. Qual é o escopo do método?
    - Quão extensível ele é? Seria simples aumentar sua escala com uma base de conhecimento maior?
    - Como ele aborda a tarefa? E partes dela? E uma classe de tarefas?
    - Ele ou suas partes poderiam ser aplicados a outros problemas?
    - Ele é transferível para tarefas mais complicadas (talvez com mais conhecimento ou mais ou menos limitações ou com interações complexas)?
6. Quando o método não consegue dar uma boa resposta, ele se abstém, dá uma resposta ruim ou dá a melhor resposta possível dados os recursos disponíveis?
7. Quão bem esse método é compreendido?
    - Por que ele funciona?
    - Em quais circunstâncias ele não funciona?
    - As limitações são inerentes ao modelos ou simplesmente não foram tratadas?
    - As decisões de *design* estão bem justificadas?
8. Qual a relação entre o problema e o método? Por que ele funciona para esta tarefa específica?

Critérios para avaliação da implementação do método

1. Quão demonstrativo é o programa?
    - Podemos avaliar seu comportamento externo?
    - Quão transparente ele é? Podemos avaliar seu comportamento interno?
    - A classe de habilidades necessárias para a tarefa pode ser demonstrada por um conjunto bem definido de casos de teste?
    - Quantos casos de teste ele demonstra?
2. Ele é refinado especialmente para um exemplo em particular?
3. Quão bem o programa implementa o método?
    - Você consegue determinar as limitações do programa?
    - Houve partes que foram deixadas de fora ou foram feitos quebra galhos? Por que e com qual efeito?

   - A implementação forçou uma definição detalhada ou mesmo uma re-avaliação do método? Como essa re-avaliação foi feita?
4. A performance do programa é previsível?

Critérios para avaliação do *design* do experimento

1. Quantos exemplos podem ser demonstrados?
   - Eles são qualitativamente diferentes?
   - Esses exemplos ilustram todas a habilidades propostas? Eles ilustram limitações?
   - O número de exemplos é suficiente para justificar generalizações indutivas?
2. A performance do programa deve ser comparada com um *standard*, como um outro programa, *experts* ou novatos, com com sua própria performance refinada? O *standard* deve ser normativo, ter validade cognitiva, basear-se em eventos do mundo real ou em simulações?
3. Quais são os critérios de uma boa performance? Quem os define?
4. O programa pode ser generalizado (ou seja, é independente do domínio)?
   - Ele pode ser testado em diversos domínios?
   - Os domínios são qualitativamente diferentes?
   - Eles representam uma classe de domínios?
   - A performance no domínio inicial deve ser comparada à performance em outros domínios? Você espera que o programa está customizado para ter uma performance melhor no(s) domínio(s) usado(s) para *debugging*?
   - O conjunto de domínios é suficiente para justificar uma generalização indutiva?
5. Uma série de programas relacionados está sendo avaliada?
   - Você consegue determinar como as diferenças entre os programas se manifestam em diferentes comportamentos?
   - Se o método foi implementado de forma distinta em cada programa da série, como essas diferenças afetam as generalizações?
   - Houve dificuldades na implementação do método em outros programas?

Critérios para avaliação dos achados do experimento

1. Qual foi a performance do programa em comparação com o *standard* selecionado (por exemplo, outros programas, humanos, comportamento normativo)?
2. A performance do programa foi diferente das predições de como o método deveria funcionar?
3. Quão eficiente é o programa em termos de requerimentos de memória e de conhecimento?
4. O programa demonstrou boa performance?

5. Você aprendeu o que você queria com o programa e os experimentos?
6. É fácil para os usuários entenderem-no?
7. Você consegue definir as limitações de performance do programa?
8. Você entende por que o programa funciona ou não funciona?
    - Qual o impacto de mudanças sutis no programa?
    - Sua performance está de acordo com o esperado em exemplos que não foram usados para *debugging*?
    - O efeito de diferentes estratégias de controle pode ser determinado?
    - Como o programa responde se o *input* é rearranjado, tem ruído ou está incompleto?
    - Qual a relação entre as características dos problemas de teste e da performance (tanto externos, ou internos caso os registros do programa estejam disponíveis)?
    - A compreensão do programa pode ser generalizada para o método? Para características do método? Para uma tarefa maior?

Referências

COHEN, P. R.; HOWE, A. E. How Evaluation Guides AI Research: The Message Still Counts More than the Medium. **AI Magazine**, v. 9, n. 4, p. 35, 1988.