

Uso de algoritmos de similaridade para classificar códigos de acordo com a taxonomia SOLO em disciplinas de programação introdutória

Alexandre^{1,2}
Barbosa

Danilo¹
Costa

Allan¹
Correia

Douglas¹
Moura

Evandro^{1,2}
Costa

¹Universidade Federal de Alagoas (UFAL)

²Universidade Federal de Campina Grande (UFCG)

alexandre.barbosa@arapiraca.ufal.br

Introdução

Contexto

- A programação é uma das competências fundamentais na área da computação.
- Disciplinas possuem um alto índice de reprovações e desistências
 - taxa média de reprovação 32,6% [Ramos et al.]
 - reprovações em torno de 50% [Barbosa et al. 2014] e [Noschang et al. 2014]
- Motivações...
 - dificuldade do professor em fornecer um acompanhamento adequado para cada um dos alunos [?]

Introdução

Contexto

- Atividades práticas de codificação são uma das estratégias utilizadas nas disciplinas de programação
- Avaliação das soluções propostas não é trivial:
 - tempo;
 - viés;
 - erros na correção;
 - critérios (saídas adequadas, estrutura do código, documentação, eficiência, manutenibilidade, ...)

Introdução

Objetivos

- Mecanismos automáticos para análise de códigos [?], [Pelz et al. 2012], [Silva et al.] e [Raabe et al. 2015]
- Neste trabalho:
 - Algoritmos de avaliação de similaridade
 - Taxonomia SOLO (*Structure of the Observed Learning Outcomes*)
 - Comparar classificação de especialistas vs. classificação obtida com os algoritmos

Metodologia

Perguntas de pesquisa e hipóteses

QP1 - Utilizando os avaliadores de similaridade, a classificação fornecida é semelhante a classificação de um especialista?

- **Hipótese nula:** Os algoritmos fornecem uma classificação com alta concordância em relação aos especialistas;
- **Hipótese alternativa:** Os algoritmos fornecem uma classificação com baixa concordância em relação aos especialistas;

QP2 - A quantidade de soluções de referência adotadas para identificar o grau de similaridade tem influência na concordância entre as classificações?

- **Hipótese nula:** A concordância na classificação não difere independente da quantidade de soluções de referência adotada;
- **Hipótese alternativa:** Quanto maior seja a quantidade de soluções de referência, melhor será a concordância na classificação;

Dados:

- 4 enunciados de exercícios práticos
- 438 soluções de alunos 1º)131 2º)118 3º)108 4º)81 - Python 3
- 16 soluções de referência (relational, multistructural, unistructural e prestructural)
- Conjunto de classificações de 3 especialistas

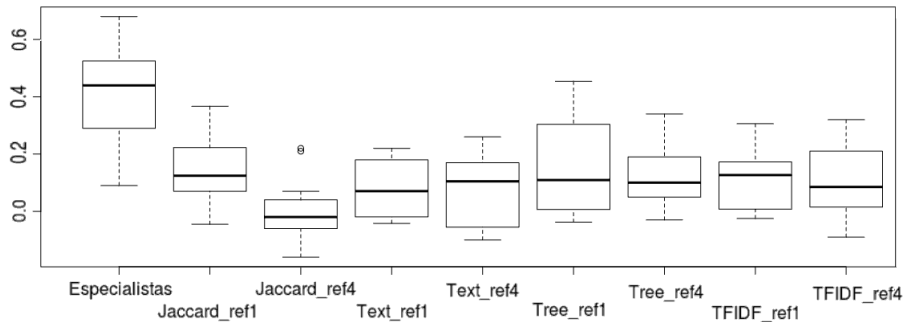
Execução:

- O experimento foi realizado em duas rodadas
 - 1ª) 1 solução (relational) - intervalos $R(0.75, 1.00]$, $M(0.50, 0.75]$, $U(0.25, 0.50]$ e $P(0.0, 0.25]$
 - 2ª) 4 soluções (relational, multistructural, unistructural e prestructural) - similaridade máxima
- Computar Kappa e Distância Euclidiana para classificações de 3 especialistas vs. classificações dos algoritmos

Resultados e Discussão

Resultados observando Kappa de Cohen

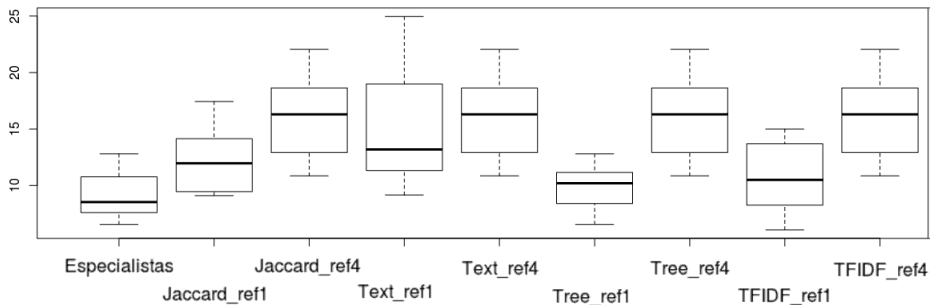
Figura: Resultados da medida de comparação Kappa de Cohen.



Resultados e Discussão

Resultados observando Distância Euclidiana

Figura: Resultados da medida de comparação Distância Euclidiana.



Resultados e Discussão

Teste de hipóteses

Tabela: Resultados do teste de Mann-Whitney-Wilcoxon* para as hipóteses.

	p-valor ($\alpha = 0.05$) (hipóteses para QP1)	p-valor ($\alpha = 0.05$) (hipóteses para QP2)
Jaccard	0.009813	6.157e-09
Text	2.2e-16	3.661e-16
Tree	0.04466	0.1887
TF-IDF	2.2e-16	0.9999

*Teste de normalidade de Shapiro-Wilk, dados não possuem distribuição normal

Conclusões e Trabalhos Futuros

Conclusões

- **Para QP1 a hipótese nula pode ser rejeitada**
- Os especialistas não adotam uma estratégia de comparação que se baseie em apenas uma solução de referência
- **Para QP2:**
 - é possível rejeitar a hipótese nula para dois algoritmos (Jaccard e Text), a classificação fornecida pelos algoritmos é **diferente** com a variação na quantidade de soluções
 - não é possível rejeitar a hipótese nula para dois algoritmos (Tree e TFIDF), a classificação fornecida pelos algoritmos não é **diferente** com a variação na quantidade de soluções
- Era de se esperar que a classificação melhorasse, pois um universo maior de códigos aceitáveis estava sendo fornecido

Conclusões e Trabalhos Futuros

Trabalhos Futuros

Trabalhos Futuros

- **Investigação:** adoção de uma maior quantidade de soluções de referência *relational*
- **Investigação:** combinação com outras estratégias

Replicação do experimento:

<https://github.com/nemo-research/walgprog-2016>

Referências



Barbosa, A. d. A., Ferreira, D. Í., and Costa, E. B. (2014).
Influência da linguagem no ensino introdutório de programação.
In Anais do SBIE, pages 612–621.



Ramos, V., Freitas, M., et al.
A comparação da realidade mundial do ensino de programação para iniciantes com a realidade nacional: Revisão sistemática da literatura em eventos brasileiros.



Noschang, L. F., Fillipi Pelz, E. A., and Raabe, A. L. (2014).
Portugol studio: Uma ide para iniciantes em programação.
In Anais do CSBC/WEI, pages 535–545.



Pelz, F. D., de Jesus, E. A., and Raabe, A. L. (2012).
Um mecanismo para correção automática de exercícios práticos de programação introdutória.
In Anais do SBIE.



Silva, M. T., Costa, E. D. B., et al.
Um Arcabouço para Construção de Mecanismos de Análise de Códigos de Programação Introdutória.



Raabe, A., de Jesus, E. A., Hodecker, A., and Pelz, F. (2015).
Avaliação do feedback gerado por um corretor automático de algoritmos.
In Anais do SBIE, page 358.

5º Congresso Brasileiro de Informática na Educação (CBIE)
2º Workshop de Ensino em Pensamento Computacional, Algoritmos e
Programação (WAlgProg)
Uberlândia/MG, 24 a 27 de Outubro de 2016

Obrigado pela atenção!

Alexandre de A. Barbosa
alexandre.barbosa@arapiraca.ufal.br