Análise de colaboração em desenvolvimento global de software

Vitor A. C. Horta[1],Victor Ströele[1], Jonice Oliveira[2], Regina Braga[1], José Maria David[1], Fernanda Campos[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Juiz de Fora (UFJF)
Caixa Postal 422, 36001-970 – Juiz de Fora – MG – Brazil

[2]COPPE/UFRJ - Computer Science Department - Graduate School and Research in Engineering – Federal University of Rio de Janeiro

Abstract. *The global open source software development popularity motivates the search for experts with capability of helping other developers in solving complex tasks. The challenge is: given a task, how to identify an expert (or set of experts) to execute it? This problem is named the expert-location problem. Some of these search difficulties are the large amount of data, the lack of technical details about the candidates and the different levels of collaboration. This work aims to detect experts and to identify groups with experienced members in some topics in Q&A forums. To achieve these goals the StackOverflow forum was used and modeled as a complex network. The presented method uses NetSCAN algorithm to detect overlapping communities in social networks. Through a temporal analysis the developers' skills were revealed. It was also found out that some users are changing their interests over time. The evaluation was conducted through a viability analysis by comparing the scores of the answers given by experts (indicated by the proposed method) and by common users.*

Resumo. *A popularidade do desenvolvimento global de software Open Source aumenta a necessidade de busca por especialistas capazes de auxiliar outros desenvolvedores na resolução de tarefas complexas. O desafio é: dada uma tarefa, como identificar o melhor especialista (ou um conjunto de especialistas) para executá-la? Este problema é chamado de localização de especialistas. Algumas das dificuldades desta pesquisa são o grande volume de dados produzido, a ausência de maiores detalhes técnicos dos envolvidos e os diferentes níveis de colaboração. O objetivo deste trabalho é detectar especialistas e identificar grupos com participantes experientes em determinados tópicos em um fórum Q&A. Para tal, o fórum StackOverflow foi modelado como uma rede complexa. O método apresentado é composto pela detecção de comunidades sobrepostas através do algoritmo NetSCAN. Através de uma análise temporal foram reveladas as aptidões dos desenvolvedores, mostrando também uma tendência de mudança de seus interesses. A avaliação do método foi feita através de uma análise de viabilidade, comparando as notas das respostas dos especialistas (apontados pelo método proposto) com as notas das respostas dos usuários comuns.*

1. Introdução

O aumento da demanda por software e o crescimento do desenvolvimento global de software despertam o interesse na busca por desenvolvedores experientes ou especialistas em determinados tópicos de desenvolvimento [Ma et al. 2009].

Esta busca por especialistas procura solucionar problemas como: roteamento de perguntas [Li and King 2010], correção de defeitos [Zhang and Lee 2012] e revisão de código fonte [Rahman et al. 2016]. O problema de roteamento de perguntas está relacionado à baixa taxa de respostas em fóruns Q&A e, uma forma de resolvê-lo, é encaminhar as perguntas para usuários que possuem maior chance de dar uma resposta relevante. A recomendação de desenvolvedores para correção de defeitos é outro problema que pode ser apoiado pela busca por especialistas. Segundo [Zhang and Lee 2012] o aumento do tamanho e da complexidade de software faz com que o número de *bugs* relatados em sistemas *open source* seja muito elevado. Uma forma de aumentar a taxa de correção destes defeitos é através da recomendação de desenvolvedores apropriados para a tarefa. A recomendação de desenvolvedores para processo de revisão de código também é outro problema encontrado em ambientes colaborativos. Essa atividade pode ser apoiada através da disponibilização de informações sobre as especialidades de desenvolvedores [Rahman et al. 2016].

Uma forma de identificar tais desenvolvedores é através da análise de fóruns Q&A (perguntas e respostas), tais como: Yahoo! answers, Quora e StackOverflow. Dentre eles, o StackOverflow se destaca por possuir uma grande quantidade de usuários, uma alta taxa de respostas e um pequeno tempo de resposta [Mamykina et al. 2011].

O StackOverflow possui uma grande quantidade de usuários ativos, mais de 5 milhões desde 2008 [Bayati 2016]. Promove a colaboração e troca de conhecimento entre essas pessoas através de perguntas e respostas no website. Algumas características importantes deste fórum são: a utilização de tags (palavras-chave) para determinar o domínio de cada conversação; e a pontuação de perguntas e respostas, que determinam a relevância de cada contribuição.

A identificação de desenvolvedores especialistas através de fóruns Q&A é um assunto que vem sendo abordado por diversos trabalhos [Li and King 2010] [Yang and Manandhar 2014] [Fu et al. 2017]. No entanto, muitos destes trabalhos propõem rankings de desenvolvedores por tópicos e não consideram elementos de colaboração em seus estudos.

Além disso, segundo Rubin [Rubin and Rinard 2016], fatores como diferença de fuso horário e de linguagem possuem grande impacto e podem reduzir a produtividade no trabalho colaborativo. Pessoas que já possuem colaborações em conjunto tendem a ser menos impactadas com estes problemas [Aggarwal 2011]. Desta forma, torna-se interessante identificar grupos de pessoas que obtiveram sucesso por colaborarem em alguma atividade. Entretanto, em função do grande volume de dados existentes nessas bases de fóruns Q&A, é problemático identificar e analisar as relações entre esses desenvolvedores, bem como identificar grupos de desenvolvedores que já tenham colaborado com sucesso em alguma atividade.

Neste sentido, este trabalho tem como objetivo identificar: (i) grupos de desenvolvedores que já tenham trabalhado colaborativamente em desenvolvimento de software, e (ii) desenvolvedores especialistas em determinados tópicos de desenvolvimento. Com isso, pretende-se identificar especialistas para resolução de problemas e pessoas capazes de trabalhar colaborativamente.

Para alcançar os objetivos deste trabalho, foi modelada uma rede de colaboração

entre desenvolvedores a partir das interações entre os usuários no StackOverflow. Após a construção dessa rede foi utilizado um método de detecção de comunidades e identificação de pessoas influentes em redes complexas, denominado NetSCAN [Vitor Horta 2017]. Com base nos resultados obtidos pelo algoritmo, foi feita uma análise detalhada da rede com o intuito de caracterizar a colaboração nos grupos identificados, e as especialidades dos desenvolvedores.

Este trabalho está organizado da seguinte forma: a seção 2 apresenta métodos para detecção de comunidades e desenvolvedores especialistas, a seção 3 detalha a modelagem da rede StackOverflow. A seção 4 discute o uso do algoritmo NetSCAN. As seções 5 e 6 mostram, respectivamente, os resultados obtidos e a avaliação dos mesmos, e na seção 7 são apresentadas as considerações finais e os trabalhos futuros.

2. Grupos de Colaboração no StackOverflow

No StackOverflow os usuários podem criar perguntas e atribuir até 5 tags a cada pergunta. Essas tags ajudam a definir o domínio da pergunta e o interesse dos usuários. Alguns exemplos de tags existentes são *"c" "java"*, *"javascript"*, *"sql"* e *"html"*. Para manter a consistência das tags o StackOverflow utiliza um sistema de recomendação de tags já existentes, além de permitir que apenas usuários mais experientes possam criar novas tags. Após a criação da pergunta os outros usuários do fórum podem então submeter uma resposta. Os próprios usuários do fórum podem votar positivamente ou negativamente nas perguntas e respostas indicando quais são as mais relevantes e com maior contribuição.

Uma característica deste tipo de fórum é que qualquer usuário pode responder e votar nas perguntas e respostas sem que haja um relacionamento pré-estabelecido entre eles. De acordo com [Meng et al. 2015] isto faz com que a estrutura de fóruns Q&A seja mais parecida com estruturas de estrela ao invés de estrutura de triângulos. Dessa forma, as pessoas formam grupos por tópicos de interesse e, por possuírem múltiplos interesses, estas pessoas participam de diferentes grupos com interesses distintos.

A identificação destes grupos pode ser alcançada através de métodos de detecção de comunidades. Para isso existem 3 principais tipos de abordagem [Meng et al. 2014]: métodos baseados em grafos, métodos de agrupamento e métodos baseados em modelos. A primeira consiste em inferir um grafo através das interações no fórum e, posteriormente, utiliza-se algum método de detecção de comunidades em redes [Xie et al. 2013][Cuijuan Wang and Wang 2015] no grafo inferido. Esta abordagem possui algumas limitações. Primeiramente, ela não utiliza os atributos dos nós e dos relacionamentos. Também não é possível saber os tópicos em que as pessoas interagiram e, portanto, não é possível identificar os tópicos de interesse dos usuários.

Outra forma de abordar este problema é através da utilização de métodos de agrupamento. Neste caso, calcula-se a similaridade dos perfis dos usuários e utiliza-se métodos de agrupamento baseados em similaridade [Meng et al. 2014]. Nesta abordagem a estrutura da rede não é considerada e cada usuário é atribuído a apenas um grupo. Esta é uma grande limitação no contexto de fóruns Q&A, já que os usuários, em geral, participam de múltiplos grupos com diferentes interesses.

Uma terceira forma de identificar grupos neste contexto é através de um modelo que considere tanto a estrutura da rede quanto os atributos dos nós e dos relacionamentos.

Esta abordagem é utilizada por [Meng et al. 2014] e [Kianian et al. 2017] e permite identificar os tópicos de interesse dos usuários, associar usuários a múltiplos grupos e detectar comunidades sobrepostas.

O modelo proposto neste trabalho se assemelha com esta terceira abordagem, mas, diferente dos modelos anteriores [Meng et al. 2014][Kianian et al. 2017], este considera que as relações entre os usuários são direcionadas e indicam a relevância das contribuições de um usuário para o outro. Assim é possível identificar quais são os usuários que fizeram as contribuições mais relevantes na rede permitindo a detecção de usuários especialistas e influentes em grupos de interesse. Outra característica deste trabalho é permitir que existam múltiplos grupos referentes a um mesmo tópico de interesse. Dessa forma, os grupos detectados além de indicarem usuários que compartilham o mesmo interesse também mostram usuários com maior potencial para futuras contribuições. Para tal foi utilizado o NetSCAN, um algoritmo para detecção de comunidades e usuários influentes em redes de grande porte.

3. Rede de Colaboração entre Desenvolvedores

A rede de desenvolvedores proposta neste trabalho foi modelada a partir das perguntas e respostas do StackOverflow. Os vértices da rede representam os usuários do fórum. As arestas representam todas as respostas dadas pelo vértice de origem para o vértice de destino em uma determinada tag.

Cada usuário pode responder várias vezes a uma mesma pergunta e, além disso, cada uma das respostas pode conter até 5 *tags*. Essas respostas recebem, cada uma delas, um *score* atribuído pelos usuários do StackOverflow.

Dessa forma, a rede de colaboração foi representada por um grafo direcionado $G = (V, E)$, onde $V = \{v_0, v_1, ..., v_{n-1}\}$ é o conjunto de *n* vértices (usuários), E representa o conjunto de arestas na forma $e_{ijt} = (v_i, v_j, t)$ entre os usuários v_i e v_j em uma determinada tag t. Como um usuário pode dar várias respostas contendo uma mesma *tag*, $R_{ijt} = \{r^0_{ijt}, r^1_{ijt}, ..., r^{l-1}_{ijt}\}$ é um conjunto de *l* respostas sobre a *tag* t entre os usuários v_i e v_j, sendo que r^k, para $0 \leq k < l$, possui o *score* dado pelos usuários à essa resposta.

Com o intuito de mensurar as contribuições entre usuários em cada *tag*, as arestas possuem um peso que representa a relevância de todas as contribuições que o usuário v_i realizou sobre v_j em uma *tag* t, definido por $IP(e_{ijt})$.

Para calcular o peso da aresta $IP(e_{ijt})$ obtém-se primeiramente o somatório dos *scores* (pontuação) de todas as respostas do usuário v_i para v_j na tag t. A Equação 1 define o cálculo deste somatório, onde $S(r^k_{ijt})$ é o *score* obtido pela k-$ésima$ resposta dada por v_i a v_j em uma tag t e l é a quantidade de respostas dadas de v_i para v_j sobre a tag t.

$$Sum_{ijt} = \sum_{k=0}^{l-1} S(r^k_{ijt}) \tag{1}$$

A Equação 1 é normalizada dividindo o somatório dos *scores* Sum_{ijt} pelo total de contribuições recebidas por v_j na tag t. A Equação 2 define o cálculo do peso da aresta $IP(e_{i,j,t})$ onde $\parallel N(j,t) \parallel$ é a soma de todos *scores* recebidos por v_j na tag t.

$$IP(e_{ijt}) = \frac{Sum_{ijt}}{\parallel N(j,t) \parallel} \quad (2)$$

Como $Sum_{ijt} \leq \parallel N(j,t) \parallel$ então $0 \leq IP(e_{ijt}) \leq 1$. Desta forma, quanto mais $IP(e_{ijt})$ se aproxima de 1 maior a relevância das contribuições de v_i para v_j na tag t. Por outro lado, quanto mais $IP(e_{ijt})$ se aproxima de 0, menor será a relevância destas contribuições. Se $IP(e_{ijt})$ = 1 então v_i foi o único usuário a contribuir positivamente com v_j nesta tag. O valor de $IP(e_{ijt})$ também pode ser 0 caso v_i tenha contribuído com v_j mas a soma dos *scores* de suas contribuições são menores ou iguais a 0.

A Figura 1 mostra uma abstração desta rede onde o usuário v_i contribuiu com o usuário v_j em m *tags* distintas. É possível que v_i tenha dado múltiplas respostas para cada *tag*, já que cada aresta representa todas as contribuições em uma mesma *tag*. Apesar de não aparecerem na figura, a quantidade de respostas e o *score* total obtido por elas estão armazenados e podem ser acessados através dos atributos de cada aresta.

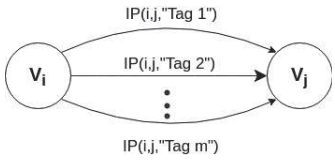

Figura 1. Interações entre dois desenvolvedores

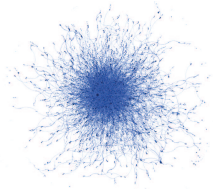

Figura 2. Visualização da rede oferecida pelo gephi

Para implementar a rede foi utilizado um *dataset* contendo dados do ano de 2008 até 2016 com 565680 usuários, 618726 perguntas (*posts*) e 1188765 respostas sobre as 20 *tags* mais frequentes do StackOverflow. A rede de desenvolvedores foi implementada através de um banco de dados orientado a grafos Neo4j (https://neo4j.com/). A Figura 2 mostra uma visualização total da rede feita com o *Gephi* (https://gephi.org/).

Como observado na Figura 2 não é viável fazer uma análise visual desta rede devido ao grande volume de dados. Neste sentido, o algoritmo NetSCAN foi utilizado para a detecção automática de grupos colaborativos e desenvolvedores especialistas.

4. Uso do NetSCAN para detecção de comunidades

Após modelar a rede e implementá-la no Neo4j foi definida uma estratégia para particionar o grafo que representa a rede e executar o algoritmo NetSCAN [Vitor Horta 2017] para detecção de comunidades em cada uma dessas partições.

O NetSCAN é um algoritmo de detecção de comunidades baseado em densidade que identifica vértices influentes (*cores*) na rede e detecta comunidades sobrepostas a partir destes *cores*. Além disso, o NetSCAN determina o número de grupos automaticamente e não limita o número de grupos que um vértice pode participar. Estas características são importantes no contexto de fóruns Q&A já que os usuários tendem a participar de vários grupos e que existem usuários especialistas em determinados tópicos de interesse.

A Figura 3 mostra um diagrama que ilustra o processo de particionamento e execução do algoritmo. O processo se inicia no passo 1 onde ocorre a implementação

da rede no Neo4j. No passo 2 uma *tag* t é selecionada e inicia-se um processo iterativo para o particionamento e execução do NetSCAN. No passo 3 é extraído o subgrafo $G_t \subset G$ que contém apenas os vértices e arestas relacionados a *tag* t. O subgrafo G_t é utilizado como entrada para a execução do NetSCAN no passo 4, onde são detectadas as comunidades e os usuários especialistas relacionados a *tag* t. O passo 5 é utilizado para decidir se novas iterações são necessárias e, caso não existam outras *tags*, o processo se encerra no passo 6.

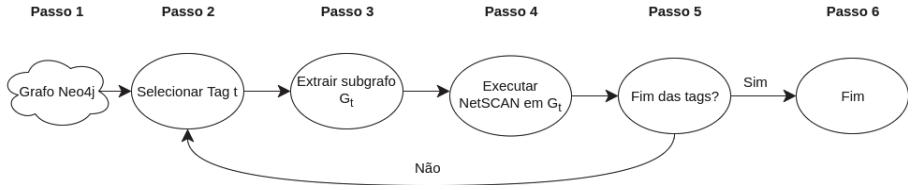

Figura 3. Diagrama para execução do algoritmo de detecção de comunidades

Ao final do processo definido na Figura 3 as comunidades detectadas são armazenadas no Neo4j por vértices do tipo *Cluster* e os usuários especialistas são os vértices do tipo *User* que possuem o atributo *core*. A próxima seção apresenta e analisa os resultados obtidos através deste processo.

5. Resultados

Nesta seção é apresentada uma análise dos resultados do processo de detecção de comunidades na rede do StackOveflow. Primeiramente foi feita uma análise exploratória dos resultados com o objetivo de levantar as principais características das comunidades e dos desenvolvedores especialistas detectados. Depois foi feita uma análise detalhada para entender o motivo real destas características.

5.1. Análise das Comunidades e desenvolvedores especialistas

Os resultados do processo de detecção de comunidades foram analisados com apoio de ferramentas de visualização e de consultas no Neo4j. As Figuras 4 e 5 mostram como são representadas as comunidades e seus membros no banco de dados.

A Figura 4 mostra uma comunidade de *python* (retângulo vermelho) com apenas um desenvolvedor especialista (vértice verde com seta), representado pelo vértice *core* central da figura. Os outros vértices azuis representam desenvolvedores que possuem interesse e participam de *posts* sobre *python*, mas não são considerados especialistas neste assunto. Assim como esperado essa comunidade possui estrutura de estrela na qual todas as ligações estão centralizadas no vértice *core*. A Figura 5 mostra uma comunidade de *c++* que contém dois desenvolvedores especialistas e que também possui estrutura de estrela. Neste caso os dois vértices *cores* desta comunidade possuem contribuições relevantes com outros usuários e também entre si.

Através das execuções do NetSCAN foram detectadas ao todo 10897 comunidades e 11996 desenvolvedores especialistas sendo que *"javascript"* foi a *tag* com maior número de comunidades (1043) e *"java"* a que possui o maior número de especialistas (1551).

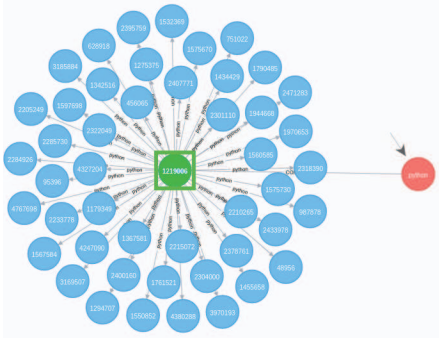

Figura 4. Comunidade (vértice vermelho com seta) de python com apenas um *core* (retângulo azul centralizado)

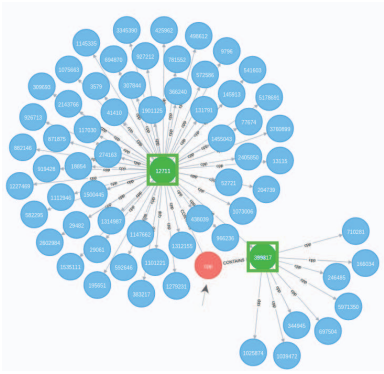

Figura 5. Comunidade de c++ com dois *cores*

Nota-se que o número total de especialistas é maior que o número total de comunidades. Isso acontece porque as comunidades podem possuir múltiplos especialistas que colaboraram entre si. Além disso, existem casos onde os especialistas participam de mais de uma comunidade, gerando sobreposições entre essas comunidades. A Figura 6 ilustra três casos de comunidades (vértices vermelhos) de diferentes tópicos de interesse que se sobrepõem através de um desenvolvedor especialista (vértice azul).

Figura 6. Sobreposição em comunidades de diferentes interesses

Estas sobreposições indicam que o desenvolvedor compartilhado entre as duas comunidades possui interesse e habilidades em ambas tecnologias, já que, para ser considerado um vértice *core* em múltiplas comunidades este usuário deve possuir colaborações relevantes nestes múltiplos assuntos.

Outra informação relevante que pode ser encontrada nas sobreposições entre as comunidades é a correlação entre dois assuntos, pois comunidades de assuntos distintos que compartilham múltiplos membros podem indicar uma alta correlação entre estes assuntos. A Tabela 1 mostra o número de usuários compartilhados entre comunidades de diferentes *tags*. Percebe-se através dessa tabela que os pares de *tags* com maior número de usuários compartilhados são de fato tecnologias muito relacionadas, sendo elas: *javascript* e jquery; *html* e *css* e; *android* e *java*. Pode-se dizer então que há indícios que de fato a quantidade de usuários compartilhados entre as comunidades indicam a correlação entre os interesses destas comunidades.

Tabela 1. Usuários compartilhados por comunidades de diferentes tags

Tag 1	Tag 2	Usuários compartilhados
”javascript”	”jquery”	9979
”html”	”javascript”	4804
”css”	”html”	4389
”android”	”java”	2476
...	...	...
”python”	”rubyonrails”	1
”iphone”	”mysql”	2

5.2. Análise temporal das sobreposições

Após perceber a existência de desenvolvedores especialistas que participam simultaneamente de múltiplas comunidades com diferentes tópicos de interesse, foi feita uma análise temporal sobre estes usuários para entender o motivo real destas sobreposições.

Nesta análise foram coletados os anos das respostas dadas pelos usuários sobrepostos em cada uma de suas comunidades. Foram detectados três principais casos que podem incentivar o surgimento das sobreposições: (i) desenvolvedor ativo em comunidades de assuntos muito relacionados; (ii) desenvolvedor ativo em comunidades de assuntos concorrentes e; (iii) desenvolvedor em processo de mudança de tópico de interesse.

O primeiro motivo é o mais comum e recorrente. Como as comunidades são caracterizadas por terem interesses em tecnologias muito relacionadas, os desenvolvedores especialistas em ambas tecnologias conseguem se manter ativos nas múltiplas comunidades. Para ilustrar este caso foi escolhido um usuário que participa ativamente de uma comunidade de *javascript* e uma comunidade de *jquery*. Percebe-se através da Figura 7 que o usuário analisado manteve sua atividade nas duas comunidades durante todo seu período de contribuição no fórum.

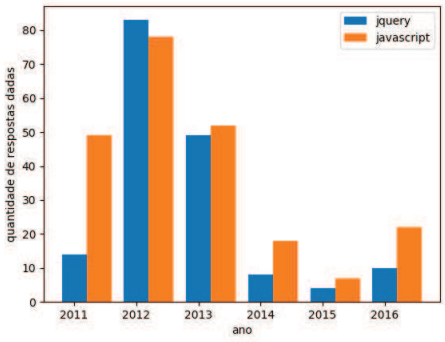

Figura 7. Histórico de atividade de um desenvolvedor especialista em comunidades de jquery e *javascript*

Outro motivo real de sobreposição encontrado acontece quando um usuário consegue se manter ativo em duas comunidades que possuem interesse em tecnologias concorrentes. Este caso é mais raro pois exige que o desenvolvedor sobreposto tenha habilidade em tecnologias pouco relacionadas ou assuntos muito distintos. A Figura 8 mostra o histórico de atividades de um usuário que se manteve ativo em uma comunidade de *android* e uma comunidade de *iOS*, mostrando assim a capacidade deste usuário em desenvolver em ambientes multiplataforma e utilizando diferentes linguagens de programação. Isto pode impactar positivamente na recomendação deste desenvolvedor, já que o mesmo pode exercer diferentes funções em um mesmo projeto.

Também é possível encontrar casos de sobreposições ocasionadas por usuários que mudaram de tópicos de interesse. Este cenário indica que o usuário envolvido já contribuiu com uma determinada comunidade no passado mas está colaborando em outros tópicos e com outras comunidades no momento atual. Este caso é mais frequente em comunidades de interesses distintos ou concorrentes, já que o desenvolvedor tende a ter mais

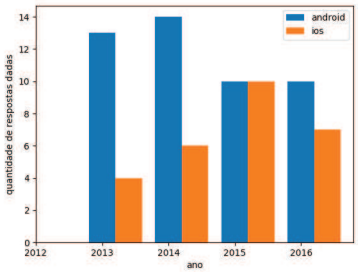

Figura 8. Atividade de um desenvolvedor especialista em comunidades de android e *iOS*

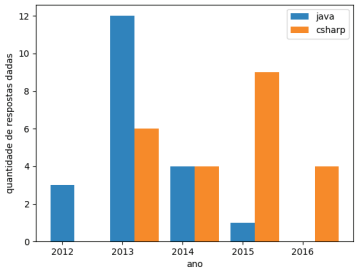

Figura 9. Transição de interesses de um desenvolvedor de java para *csharp*

dificuldade em se manter atualizado e ativo nas diferentes tecnologias e acaba migrando de comunidade.

No gráfico da Figura 9 é possível ver que o usuário esteve ativo em uma comunidade com interesse em *java* no início de suas colaborações, mas ao longo do tempo passou a contribuir com uma outra comunidade sobre *csharp*. Isto mostra que mesmo possuindo habilidades em *java* este desenvolvedor está mais apto no momento a participar de atividades e responder questões sobre *csharp*.

Foram encontrados também especialistas que participam de mais de duas comunidades, chegando a um máximo de 12 comunidades para um mesmo usuário. Estes casos porém são menos recorrentes e, dessa forma, não foi possível identificar suas principais motivações.

Os resultados desta análise temporal das sobreposições mostram que o estudo do histórico do usuário pode indicar a multidisciplinaridade deste desenvolvedor e seus interesses e aptidões atuais, podendo auxiliar em processos de tomada de decisão como a recomendação de desenvolvedores e roteamento de perguntas.

6. Avaliação

Nesta seção foi feita uma análise com dados recentes do StackOverflow para descobrir se os especialistas encontrados pelo NetSCAN são de fato pessoas ativas e com alto conhecimento nos tópicos de interesse relacionados. Para isso foi selecionado um conjunto de teste contendo as respostas dadas por usuários especialistas e não especialistas (apontados pelo método proposto) nos anos 2017 e 2018. Dessa forma nenhuma resposta deste conjunto de teste foi utilizada previamente no agrupamento realizado pelo NetSCAN.

Os usuários foram separados em dois grupos A (especialistas) e B (não especialistas), cada grupo contendo 20 usuários escolhidos aleatoriamente. Para cada usuário foi calculada a média dos scores de suas respostas mais recentes. As médias obtidas por cada grupo foram então comparadas com o intuito de descobrir se o desempenho alcançado pelos especialistas foi maior que dos não especialistas.

O *boxplot* da Figura 10 mostra o desempenho alcançado por usuários dos grupos A e B deste conjunto de teste. Pode-se perceber através do gráfico que os *scores* das respostas dadas pelos especialistas são superiores as respostas dos não especialistas. O *outlier* pertencente ao grupo A refere-se a um usuário que além de ter muitas respostas

recentes possui uma resposta com score de 262, sendo esta muito acima da média. Por outro lado os dois outliers do grupo B são usuários que tiveram uma resposta com *score* acima da média mas a maioria de suas outras respostas possuem *score* igual a 0. Além disso ambos os grupos possuem usuários com média igual a 0, que são usuários que não deram nenhuma resposta no período analisado ou não obtiveram nenhum *score* positivo em suas respostas.

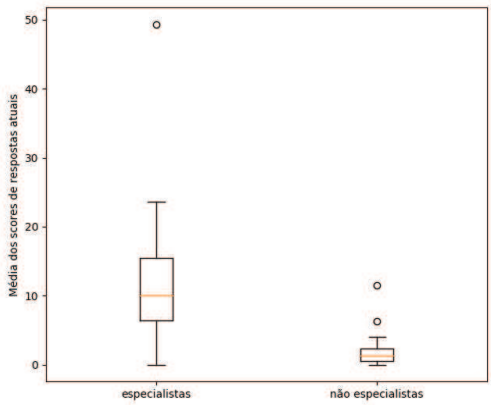

Figura 10. Desempenho dos usuários especialistas e não especialistas

Para avaliar se existe uma diferença significativa na média do desempenho dos dois grupos, os *scores* dos grupos A e B foram submetidos a um teste estatístico. Primeiramente verificou-se a não normalidade dos dados através do teste de *Kormogorov-Smirnov*. Depois foram levantadas duas hipóteses H0 (hipótese nula) e H1 (hipótese alternativa):

H0: As médias dos *scores* dos grupos A e B são iguais.

H1: As médias dos *scores* dos grupos A e B são diferentes.

Os *scores* foram então comparados pelo teste de *Mann-Whitney* com nível de confiança de 95%. Como resultado foi encontrado *p-value* $< 0,05$ e, dessa forma, rejeitou-se a hipótese nula de que as médias são iguais e aceitou-se a hipótese alternativa de que as médias são diferentes. Como a média do grupo A é maior que a do grupo B, aceitou-se que a média dos especialistas é maior que a média dos não especialistas.

A partir da análise deste conjunto de teste é possível perceber que as respostas dadas pelos usuários detectados como especialistas pelo NetSCAN possuem uma maior aceitação do que os usuários não especialistas. Este resultado aponta que o NetSCAN foi capaz de detectar tais usuários com maior *expertise* em determinados tópicos de interesse.

7. Considerações finais

Neste trabalho foi feita uma análise de redes sociais sobre o fórum Q&A de desenvolvimento de software StackOverflow com o objetivo de encontrar desenvolvedores especialistas e grupos colaborativos com desenvolvedores experientes. Para isso foi modelada uma rede social utilizando as respostas dadas pelos usuários do fórum. As *tags* de cada resposta foram utilizadas para definir o contexto de cada conversação e o *score* foi usado para medir a relevância das contribuições entre os usuários.

Para identificar grupos de desenvolvedores e usuários especialistas em cada tópico de interesse foi utilizado o algoritmo NetSCAN para detecção de comunidades e usuários influentes em redes sociais. Em um processo iterativo o grafo que representa a rede foi particionado em subgrafos relacionados a cada *tag* existente na rede e então o NetSCAN foi executado em cada um destes subgrafos.

Como resultados deste processo foram identificados grupos de desenvolvedores que colaboraram entre si em determinados tópicos de interesse e usuários especialistas nestes tópicos. Foram encontradas comunidades com múltiplos participantes influentes e comunidades sobrepostas com diferentes interesses. Em uma análise temporal identificou-se a existência de desenvolvedores multidisciplinares que atuam simultaneamente com tecnologias semelhantes e outros que atuam com tecnologias concorrentes. Também foram encontrados casos de usuários que começaram sua atividade no fórum colaborando sobre um tópico de interesse e depois migraram para outro tópico de interesse concorrente. Com base nestes resultados, os desenvolvedores especialistas podem ser recomendados para realização ou colaboração em tarefas complexas em desenvolvimento global de software.

A avaliação da proposta deste trabalho foi feita através da comparação do desempenho dos usuários especialistas detectados pelo NetSCAN com o dos usuários não especialistas. Para isso foi coletado um conjunto de teste contendo respostas mais recentes do StackOverflow e os *scores* destas respostas foram utilizados para medir o desempenho dos usuários. Através da análise de *boxplot* e de testes estatísticos pôde-se perceber que as respostas dadas pelos especialistas possuem maior aceitação do que as respostas dos não especialistas, o que aponta para a viabilidade da solução.

Como o um conjunto de teste coletado não contemplou toda a rede utilizada, outras formas de avaliação se fazem necessárias. Assim, como trabalhos futuros pretende-se avaliar os resultados junto à comunidade através de formulários e entrevistas e utilizar um conjunto de teste maior para a avaliação.

Referências

Aggarwal, C. C., editor (2011). *Social Network Data Analytics*. Springer US.

Bayati, S. (2016). Security expert recommender in software engineering. In *Proceedings of the 38th International Conference on Software Engineering Companion*, ICSE '16, pages 719–721, New York, NY, USA. ACM.

Cuijuan Wang, Wenzhong Tang, B. S. J. F. and Wang, Y. (2015). Review on community detection algorithms in social networks. In *2015 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pages 551–555.

Fu, C., Zhou, M., Xuan, Q., and xiang Hu, H. (2017). Expert recommendation in oss projects based on knowledge embedding. *2017 International Workshop on Complex Systems and Networks (IWCSN)*, pages 149–155.

Kianian, S., Khayyambashi, M. R., and Movahhedinia, N. (2017). Fuseo: Fuzzy semantic overlapping community detection. *Journal of Intelligent Fuzzy Systems*, 32(6):3987–3998.

Li, B. and King, I. (2010). Routing questions to appropriate answerers in community question answering services. In *Proceedings of the 19th ACM International Con-*

ference on Information and Knowledge Management, CIKM '10, pages 1585–1588, New York, NY, USA. ACM.

Ma, D., Schuler, D., Zimmermann, T., and Sillito, J. (2009). Expert recommendation with usage expertise. In *2009 IEEE International Conference on Software Maintenance*, pages 535–538.

Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., and Hartmann, B. (2011). Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2857–2866, New York, NY, USA. ACM.

Meng, Z., Gandon, F., Faron Zucker, C., and Song, G. (2014). Empirical Study on Overlapping Community Detection in Question and Answer Sites. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, Beijing, China.

Meng, Z., Gandon, F., and Zucker, C. F. (2015). Simplified detection and labeling of overlapping communities of interest in question-and-answer sites. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 1, pages 107–114.

Rahman, M. M., Roy, C. K., and Collins, J. A. (2016). Correct: Code reviewer recommendation in github based on cross-project and technology experience. In *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, pages 222–231.

Rubin, J. and Rinard, M. (2016). The challenges of staying together while moving fast: An exploratory study. In *Proceedings of the 38th International Conference on Software Engineering*, ICSE '16, pages 982–993, New York, NY, USA. ACM.

Vitor Horta, Victor Ströele, Fernanda Campos. José Maria N. David. Regina Braga. (2017). Redes sociais científicas: análise topológica da influência dos pesquisadores. *Sbbd proceedings 32nd Brazilian Symposium on Databases*.

Xie, J., Kelley, S., and Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Comput. Surv.*, 45(4):43:1–43:35.

Yang, B. and Manandhar, S. (2014). Exploring user expertise and descriptive ability in community question answering. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 320–327.

Zhang, T. and Lee, B. (2012). How to recommend appropriate developers for bug fixing? In *2012 IEEE 36th Annual Computer Software and Applications Conference*, pages 170–175.