Quem é quem na lista de discussão? Identificando diferentes e-mails de um mesmo participante

José Teodoro da Silva, Marco Aurélio Gerosa
Departamento de Ciência da Computação
Universidade de São Paulo (USP)
{jteodoro, gerosa}@ime.usp.br

Igor Scaliante Wiese, Igor Steinmacher
Departamento de Ciência da Computação
Universidade Tecnológica Federal do Paraná (UTFPR)
{igor, igorfs}@utfpr.edu.br

RESUMO

Listas de discussão possibilitam a comunicação entre várias pessoas utilizando a estrutura do e-mail. Listas são utilizadas para discutir diversos assuntos, desde entretenimento até desenvolvimento de software. Elas constituem uma fonte rica de informações sobre a comunicação de seus membros e o histórico das interações é utilizado para estudos quantitativos sobre o comportamento, organização e evolução da comunidade ali existente. Entretanto, usuários utilizam múltiplos endereços de e-mail, que acabam sendo interpretados como diferentes pessoas em muitos estudos, distorcendo os resultados das análises de redes sociais e levando a conclusões equivocadas. Para evitar esse tipo de problema, alguns trabalhos propõem heurísticas para determinação única do autor das mensagens, porém pouco se sabe sobre o quão efetiva são essas heurísticas. O objetivo deste trabalho é comparar três heurísticas de desambiguação de autores utilizadas na literatura. Neste estudo, utilizamos as listas de discussão de 16 projetos de software livre da Fundação Apache e encontramos indícios de que o número de endereços de e-mails utilizados na comunidade pode influenciar a qualidade dos resultados das heurísticas e que a escolha da heurística de identificação de autores depende do conjunto de dados a ser utilizado. Os resultados deste trabalho podem servir de base para pesquisadores que investigam listas de discussão de comunidades abertas com grande número de participantes.

Keywords

Atribuição de autoria; listas de discussão; Fundação Apache; sistemas de comunicação; mineração de repositórios; desambiguação de e-mails.

1. INTRODUÇÃO

Uma lista de discussão possibilita a comunicação e a difusão de informação para os participantes utilizando a estrutura do e-mail [20]. Listas são usadas para discutir vários assuntos, desde arte e entretenimento até o desenvolvimento de software. Listas são usadas tanto por pequenos grupos quanto por grandes comunidades abertas de produção coletiva, com centenas ou milhares de participantes. Nas comunidades de software livre, por exemplo, essas listas são de grande importância, sendo utilizadas para informar sobre o status do projeto, discutir sobre problemas no software, procurar por instruções de uso, coordenar os membros do projeto, enviar avisos e normas, etc. [10].

Os históricos de listas de discussão são uma fonte rica de informações para pesquisas que exploram a comunicação e interação social [11]. Listas são utilizadas para entender a estrutura de liderança e relacionamentos entre os membros da comunidade [25] e para analisar padrões de discurso de estudantes [18], por exemplo. Técnicas de análise de redes sociais são aplicadas, extraindo de forma algorítmica os participantes (nós da rede) e as trocas de mensagens (arestas). Em projetos de desenvolvimento de software, listas de discussão são utilizadas para estudar diferentes aspectos do desenvolvimento de software, por exemplo, explorar a estrutura da comunidade [27], analisar a rede social da comunidade a partir da sua comunicação [22], entender a evolução do software livre a partir de discussões [6], estudar os papéis dos membros na lista [17], analisar seu processo e práticas de desenvolvimento [1, 10], explorar a comunicação entre seus colaboradores [16] e analisar como a participação na lista pode afetar os novos membros da comunidade [26].

Entretanto, a extração de dados por meio de algoritmos não é uma tarefa trivial. Entre outros, há o problema de identificação dos autores das mensagens, conforme relatado por Bettenburg et al. [2] e Bird et al. [4]. Falhas na identificação podem gerar atribuição incorreta das mensagens a um autor e invalidar os resultados de análises quantitativas [2]. A dificuldade na extração de dados das listas se dá pelo modo como os membros utilizam o e-mail: usuários criam endereços de e-mail relativamente curtos[1]; os participantes utilizam apelidos; não existe padronização na criação de contas de e-mail; endereços de e-mail empresariais são abandonados quando o usuário deixa a empresa; membros da comunidade usam seus endereços de e-mail pessoais e profissionais para participar na lista; e os clientes de e-mail são configurados com o nome do remetente de maneira inconsistente ou inexistente.

Para solucionar o problema da identificação dos autores das mensagens, pesquisadores como Bird et al. [4], Oliva et al. [17], Goeminne e Mens [9] e Kouters et al. [14] propuseram heurísticas que utilizam as informações existentes na própria lista de discussão para identificar múltiplos endereços de um participante. Ainda assim, muitos trabalhos da literatura não realizam um pré-processamento dos dados, considerando cada endereço de e-mail como sendo de uma pessoa distinta. Parte disso pode advir da falta de conhecimento do problema ou de seus efeitos. Há uma carência de trabalhos que avaliam e comparem a eficácia dessas heurísticas.

O objetivo deste trabalho consiste em avaliar os resultados do uso de três heurísticas para identificação de autores encontradas na literatura. Utilizamos informações publicadas pela Fundação de Software Apache para a construção de uma base de referência com endereços de e-mail dos participantes da lista de discussão de 16 projetos.

[1] Um estudo preliminar aponta indícios de que 44% dos membros dos 16 projetos avaliados neste trabalho utilizam endereços de e-mail com até sete caracteres.

Este trabalho está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 apresenta detalhes do método de construção da base de referência e da avaliação das heurísticas; a Seção 4 apresenta os resultados e discussão das avaliações realizadas; a seção 5 apresenta as ameaças à validade do estudo e a Seção 6 apresenta as conclusões e trabalhos futuros.

2. TRABALHOS RELACIONADOS

A desambiguação de informações que representam um mesmo conceito ocorre quando múltiplos formatos, ou a ausência de qualquer padrão, são utilizados para representar um mesmo atributo. Este problema é maior quando são utilizados dados de um longo período de tempo [7]. Exemplos de usos para a desambiguação são a identificação de autores em coleções de artigos científicos [7], a identificação de indivíduos por seus nomes em coleções de documentos [8] e a identificação de autores de mensagens em listas de discussão [4]. Neste trabalho, quando falarmos sobre identificação de autores estaremos nos referindo à identificação no contexto das listas de discussão.

Podemos dividir os trabalhos relacionados à identificação de autores em listas de discussão em quatro grupos: trabalhos que alertaram sobre as dificuldades de extração de informações das listas de discussão e identificação dos autores das mensagens; trabalhos que propuseram soluções para desambiguação de identidades; trabalhos que utilizaram as heurísticas e trabalhos que avaliaram as soluções propostas.

2.1 Dificuldades de extração

No primeiro grupo, trabalhos que alertam sobre as dificuldades de extração de informações das listas de discussão e sua atribuição de autoria, encontramos o trabalho de Bettenburg et al. [2]. Esse trabalho enumera os desafios existentes no processamento de dados oriundos das listas de discussão, dentre eles a atribuição de autoria e a remoção de mensagens automáticas. Os autores alertam sobre os possíveis erros de análise que ocorrem devido a uma má determinação dos autores das mensagens.

Hemmati et al. [12] também descrevem as dificuldades de recuperação de dados das listas. Eles realizaram um levantamento das boas práticas utilizadas em artigos publicados em conferências e workshops. Hemmati et al. apontam para as dificuldades existentes na recuperação das características da comunidade e apresentam várias abordagens utilizadas pelos pesquisadores na tentativa de mitigar os erros de atribuição incorreta de endereços de e-mail a participantes no processo de desambiguação de autores.

2.2 Propostas de desambiguação

Uma das abordagens mais citadas na literatura é a de Bird et al. [4], que apresenta uma heurística para recuperação desambiguação de autores da lista de discussão. Sua abordagem utiliza o agrupamento de endereços de e-mail a partir da similaridade entre os endereços e nomes encontrados nos cabeçalhos das mensagens da lista de discussão. Sua heurística mapeia padrões comuns de criação de endereços de e-mail e procura identificar similaridades entre os vários endereços de e-mail de uma mesma pessoa.

Essa heurística utiliza a similaridade de Levenshtein [15] para avaliar a semelhança entre dois nomes/endereços. Quaisquer endereços/nomes que obtiverem uma similaridade acima de um limite de tolerância (0,93) são consideradas como pertencentes à mesma pessoa. Antes de realizar a identificação, esta heurística remove acentos e pontuações nos nomes e divide o nome completo em duas partes: nome e sobrenome. Dados a função de similaridade *simil*, o *prefixoDoEmail* do endereço de e-mail (sem o domínio), o limite de tolerância *t* e o *nomeCompleto* dividido entre *nome* e *sobrenome*, a heurística considera os endereçoA e endereçoB como pertencentes à mesma pessoa nos seguintes casos:

- *simil*(*nomeCompletoA*, *nomeCompletoB*) ≥ t;
- *simil*(*nomeA*, *nomeB*) ≥ t E *simil*(*sobrenomeA*, *sobrenomeB*) ≥ t;
- *prefixoDoEmailB* contém *nomeA* e *sobrenomeA*;
- *prefixoDoEmailB* contém *nomeA* e a primeira letra do *sobrenomeA*;
- *prefixoDoEmailB* contém a primeira letra do *nomeA* e *sobrenomeA* completo; ou
- *simil*(*prefixoEmailA*, *prefixoEmailB*) ≥ t.

Uma abordagem similar à de Bird et al. [4] é utilizada por Canfora et al. [5]. O trabalho de Canfora et al. utiliza a abordagem de inicias de nomes e sobrenomes de Bird et al., mas não utiliza a similaridade entre nomes e e-mails para evitar a ocorrência de falsos positivos. Essa heurística é utilizada para analisar as características sociais dos membros da comunidade durante a correção de falhas nos projetos FreeBSD e OpenBSD [5].

Robles e Gonzalez-Barahona [24] adotam uma abordagem também utilizando nomes. Os autores consideram as possíveis combinações de nome e sobrenome para formação dos endereços de e-mail. Num segundo passo para identificação dos autores, eles buscam chaves públicas que identificam um usuário e seus e-mails. Este segundo passo limita o número de projetos para avaliação porque nem todos os projetos possuem a política de uso dessas chaves. A heurística é apresentada em um conjunto de métodos para integração de informações advindas da lista de discussão e do repositório de códigos fontes para exploração e análise de dados da comunidade Gnome.

Oliva et at. [17] adotam uma abordagem distinta. A heurística agrupa os endereços de e-mail a partir da reincidência do uso do nome do membro com seus possíveis endereços de e-mail no cabeçalho das mensagens da lista. Eles utilizam essa heurística para explorar a caracterização dos papéis dos desenvolvedores no projeto Apache Ant. Eles partem do pressuposto de que as pessoas utilizam o mesmo nome na configuração de seus clientes de e-mail, apesar de poderem utilizar endereços de e-mail distintos. Esta heurística considera que dois endereços pertencem à mesma pessoa utilizando o nome de usuário (incluindo o domínio) e o apelido utilizado na mensagem. Por exemplo, dados os quatro pares de nomes e endereços de e-mail: <José Teodoro Silva, jteodoro@usp.br>, <José, jteodoro@usp.br>, <José Teodoro de Oliveira, jteodoro@hotmail.com> e <José Teodoro de Oliveira, joliveira@meumail.org>, temos que o terceiro e o quarto pares serão igualmente identificados como pertencendo à mesma pessoa devido ao nome idêntico "José Teodoro de Oliveira". O primeiro e segundo pares serão considerados pertencentes à mesma pessoa devido ao mesmo endereço de e-mail "jteodoro@usp.br". Contudo, o primeiro e segundo pares não serão vinculados com os dois últimos porque a heurística considera os domínios do endereço de e-mail. Isso evita que a heurística atribua incorretamente todos os quatro pares à uma mesma pessoa.

Kouters [13] propõe o uso de Análise semântica latente (LSA - Latent Semantic Analysis) para identificação dos autores. Essa técnica é utilizada para determinar a similaridade entre nomes e

endereços de e-mail, identificando assim agrupamentos de endereços que potencialmente pertencem a uma mesma pessoa. Essa heurística é apresentada em sua dissertação de mestrado. Para avaliar seus resultados, Kouters conduziu um estudo de caso no projeto Gnome. Ele compara seus resultados com os de Bird et al. [4] e com um algoritmo ingênuo. Entretanto, ele utiliza uma base de referência criada a partir de três inspeções manuais do repositório de códigos fontes do projeto, não incluindo outras fontes de dados possíveis para identificação dos autores.

A heurística ingênua documentada por Kouters et al. [14] considera que dois endereços pertencem à mesma pessoa utilizando o nome de usuário (sem o domínio) e o apelido utilizado na mensagem. Considerando o mesmo exemplo dos quatro pares de endereços de e-mail e nomes, o algoritmo ingênuo vincula todos esses endereços à mesma pessoa. O primeiro, o segundo e o terceiro pares são vinculados devido ao prefixo "jteodoro" do endereço de e-mail. O terceiro e quarto pares serão vinculados devido ao nome "José Teodoro de Oliveira" em comum.

Diferentemente das heurísticas mediadas por computador, Guzzi et al. [10] realizaram a identificação de autores manualmente para um conjunto de 506 discussões da lista, perfazendo um total de 2400 mensagens. Essa abordagem se torna inviável quando o número de mensagens a serem utilizadas é muito grande. Eles a utilizaram para analisar quantitativa e qualitativamente a lista de discussão do projeto Lucene para caracterizar os assuntos tratados na lista e explorar a participação dos membros centrais do projeto nessas discussões.

2.3 Usos das heurísticas propostas

Dentre os trabalhos que utilizam as heurísticas para identificação de autores de listas de discussão, podemos citar Panichella et al. [19], que construíram redes sociais a partir das interações extraídas de várias ferramentas de comunicação (listas de discussão, fóruns do projeto e gerenciador de tarefas). Eles utilizaram a abordagem proposta por Bird com algumas modificações na tentativa de reduzir os falsos positivos. Tal abordagem também foi utilizada por Xuan e Filkov [28], construindo uma variação própria daquela proposta por Bird et al. [4] para resolver a identidade dos autores dos e-mails da lista de discussão. Eles analisam métodos quantitativos para determinação de ocorrência de atividades sincronizadas na comunidade e para explorar a produtividade e comunicação dos membros do projeto.

Rigby et al. [21] se ateve à implementação original de Bird et al. [4] e utilizou-se da mesma ferramenta criada por Bird para identificar os autores em seu estudo sobre as práticas de revisão de código no projeto httpd Apache Server. Essa mesma ferramenta foi utilizada por Nia et al. [16] para examinar a estabilidade das métricas de redes sociais quando expostas a dados ruidosos e esparsos advindos de listas de discussão. Essa ferramenta ainda foi utilizada por Bird et al. [3] para analisar a estrutura complexa de comunidades de software livre e explorar o modo como essas comunidades se auto organizam.

2.4 Avaliação das heurísticas na literatura

Dentre os trabalhos que realizam a avaliação dos resultados das heurísticas propostas, podemos citar o trabalho de Goeminne et al. [9]. Eles avaliaram os resultados de algoritmos de identificação de autores para quatro heurísticas: um algoritmo ingênuo de identificação; Bird; Robles e uma versão melhorada de Bird incluindo partes do método de Robles. Essa avaliação foi realizada para três projetos de software livre: Evince, Brasero e Subversion. Contudo, a construção de uma base de referência é limitada e realizada mediante trabalho manual. Essa base também não considera informações advindas dos sites dos projetos e do gerenciador de tarefas, que possuem informações oficiais do projeto sobre os membros.

Com exceção dos trabalhos de Kouters [13] e de Goeminne et al. [9], os pesquisadores não possuem comparação ou avaliação das heurísticas existentes na literatura de identificação de autores. Entretanto, os trabalhos de Kouters e de Goeminne et al. utilizam poucos projetos e não exploram as consequências que a mudança no tamanho do conjunto de dados pode causar.

3. MÉTODO

A extração das informações da lista de discussão pode ser realizada em qualquer projeto que disponibilize seu histórico publicamente. Entretanto, a avaliação dos resultados de identificação de autores não pode contar unicamente com as informações advindas da lista, uma vez que estamos tentando justamente validar os resultados de heurísticas que utilizem apenas as informações da lista na identificação de usuários. Por este motivo, escolhemos projetos que disponibilizam outras fontes de informações que possibilitam a avaliação dos resultados das heurísticas.

Um resumo do método de pesquisa é apresentado na Figura 1 e está discutido nas subseções a seguir.

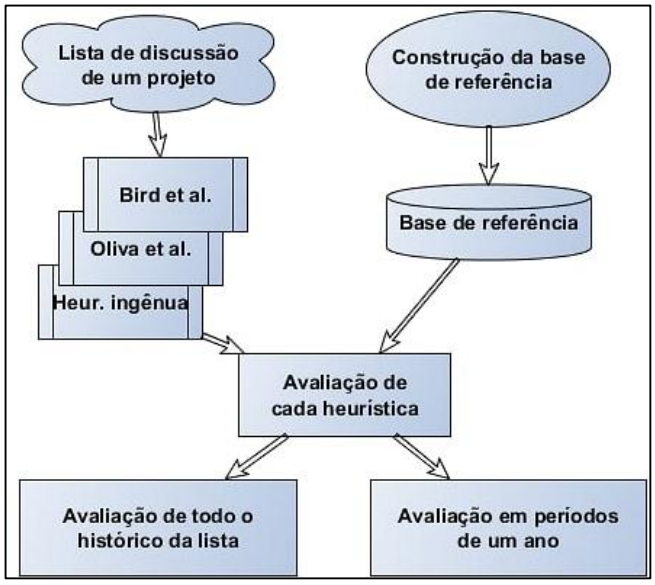

Figura 1 – Design da pesquisa

3.1 Fonte de dados

A comunidade Apache foi escolhida pela diversidade de projetos que possui e por ser frequentemente estudada empiricamente [4, 17, 26]. Além disso, a comunidade está dispersa por vários países e conta com ajuda de profissionais de diferentes empresas. A escolha dos projetos foi realizada mediante a disponibilidade das informações para construção de uma base de referência para avaliar os resultados das heurísticas.

A Fundação Apache mantém o histórico das listas de discussão de seus projetos em sua página[2]. Utilizamos os históricos das

[2] http://mail-archives.apache.org/

mensagens das listas de desenvolvedores de 16 projetos da Fundação Apache: ActiveMQ, Ant, Commons-DBCP, Commons-Collections, Cxf, Derby, Hadoop, Hbase, Httpd, Jmeter, Mahout, Maven, OpenJPA, Tomcat, TomEE e Wicket. Foram recuperados o histórico das mensagens até maio de 2013. Esses dados totalizam aproximadamente 2,5 milhões de mensagens e mais de 26 mil endereços de e-mail.

Assim como muitas comunidades de software livre, o histórico das listas da Fundação Apache são armazenados no formato mbox [23]. Esse formato contém os cabeçalhos completos dos e-mails enviados. As heurísticas utilizam essas informações para identificar os vários endereços de e-mail utilizados por um membro da comunidade.

Tal como Squire [25], utilizamos várias fontes de dados para recuperar informações e criar um conjunto de dados contendo os endereços de e-mail dos membros da comunidade. Porém, utilizamos apenas os dados existentes nos perfis de usuários encontrados no gerenciador de tarefas dos projetos (Jira) e os sites dos projetos que continham informações sobre os membros. Essas fontes foram utilizadas para a criação de uma base contendo os endereços de e-mail e nomes dos membros.

Os resultados das heurísticas podem ser expressos por agrupamentos de endereços de e-mail e nomes de usuários. Utilizamos a base de referência contendo nomes e endereços formalmente atribuídos para avaliar a capacidade destas heurísticas de realizar esse agrupamento corretamente.

3.2 Construção da base de referência de endereços

No contexto dos projetos de software livre, as ferramentas utilizadas pelos membros da comunidade podem ser utilizadas como fonte de dados confirmatórios sobre as identidades dos membros da lista de discussão. Ferramentas como gerenciador de versões, sites do projeto e gerenciador de tarefas (*Jira*) contêm informações sobre os membros que podem ser utilizadas para avaliar a qualidade dos resultados gerados pelas heurísticas.

O gerenciador de tarefas (Jira) é uma ferramenta para auxiliar no desenvolvimento e organização de projetos de software. A ferramenta possui um perfil para cada usuário, onde encontramos um endereço de e-mail, o nome e uma identificação única do usuário na comunidade.

Por sua vez, os sites dos projetos contêm informações sobre o software e sua equipe de desenvolvimento. Alguns sites disponibilizam os nomes, identificações únicas e endereços de e-mail dos membros da equipe do software. Por fim, o repositório de chaves públicas da comunidade contém uma assinatura digital acompanhada dos endereços de e-mail assinados pelo próprio membro da comunidade. Recuperamos todas essas informações que estavam disponíveis para os projetos analisados e compilamos nossa base para conduzir as avaliações.

A construção dessa base tem como objetivo recuperar o maior número de endereços de e-mail que caracterizem um mesmo usuário. Utilizamos as identificações do repositório de códigos fontes dos projetos para identificar os usuários com permissão de escrita no repositório e, posteriormente, localizar os usuários no Jira de cada projeto. A partir do perfil do usuário do Jira, fomos capazes de recuperar endereços de e-mails de 577 usuários. Além disso, realizamos uma inspeção manual nos sites dos projetos e recuperamos endereços de e-mail de 309 usuários. Do repositório de chaves públicas da comunidade, recuperamos endereços de e-mails de 723 usuários. Por fim, mesclamos as informações de todos os usuários que encontramos, utilizando a identificação única do usuário na comunidade para unir informações de um mesmo membro advindas das diferentes fontes utilizadas. Após essa fusão, nossa base de referência apresentou 933 identidades com múltiplos endereços de e-mail. Essa base contém apenas os vínculos de endereços que estão oficialmente documentados nos sites do projeto e no gerenciador de tarefas da comunidade.

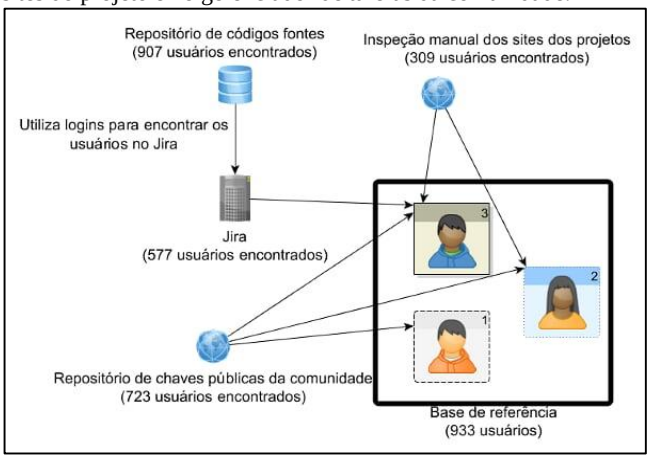

Figura 2 - Construção da base de referência utilizando várias fontes de informações

A título de exemplo, na Figura 2 podemos observar que o usuário 1 recebe endereços de e-mail apenas do repositório de chaves públicas, enquanto que o usuário 2 recebe informações advindas do servidor de chaves públicas e da inspeção manual dos sites dos projetos. Em contrapartida, o usuário 3 recebe informações de todas as fontes: Jira, repositório de chaves públicas e inspeção manual dos sites.

Apesar de nossa base de referência possuir apenas endereços de e-mail oficiais dos membros da comunidade, não é possível afirmar que todos os seus endereços estão inclusos nesta base. Os conjuntos dos endereços que recuperamos do Jira e dos sites do projeto não garantem que sabemos todos os endereços que os membros utilizaram para enviar e-mail para a lista. Neste sentido, nossa base é precisa, porém incompleta. A Tabela 1 apresenta o número de endereços de e-mail da lista, o número de endereços identificados na base de referência para cada projeto avaliado e a proporção entre esses conjuntos.

Tabela 1 - Números de endereços de e-mail utilizados na base de referência de cada projeto e encontrados nas listas.

Projeto	Endereços passíveis de comparação na base de referência para o projeto	Endereços existentes na lista	Proporção
ActiveMQ	70	966	7%
Ant	101	2857	4%
Collections	275	3061	9%
Cxf	140	679	21%
DBCP	114	1156	10%
Derby	134	760	18%
Hadoop	114	1156	10%
HBase	87	600	15%
Httpd	242	4557	5%
JMeter	19	117	16%
Mahout	57	489	12%
Maven	234	2162	11%
OpenJPA	101	464	22%
Tomcat	156	6268	2%
Tomee	22	188	12%

Wicket	51	709	7%

3.3 Avaliação das heurísticas

Os trabalhos que definem abordagens para resolução de autores de mensagens nem sempre são claros sobre todos os parâmetros envolvidos na implementação das heurísticas. Comparamos neste trabalho heurísticas para as quais obtivemos acesso ao código fonte ou que possuíam a definição suficientemente clara na literatura para possibilitar a implementação. Para evitar equívocos de implementação, entramos em contato com os autores dos trabalhos para utilizar a me sma implementação criada pelos autores, contudo apenas Bird e Oliva responderam até o momento. A comparação se dá apenas entre três métodos devido à dificuldade de conseguir as implementações dos demais trabalhos. Como trabalho futuro, realizaremos a comparação com as demais heurísticas dos autores que responderem nossa solicitação.

Realizamos a identificação dos membros da comunidade utilizando as listas de discussão para três heurísticas encontradas na literatura: a heurística utilizada por Bird et al. [4] que procura inferir os endereços de e-mail a partir da similaridade dos pares (nome, endereço de e-mail) recuperados dos cabeçalhos das mensagens; a heurística utilizada por Oliva et al. [17] que explora o uso reincidente de nomes e endereços de e-mail por um usuário e a heurística ingênua utilizada como referência por Kouters [13].

Para avaliar cada heurística, comparamos os conjuntos de endereços de e-mail identificados pela heurística contra os endereços de e-mail existentes na nossa base de referência. Essa comparação foi realizada mediante verificação dos conjuntos de endereços de e-mail do resultado de uma heurística contra o conjunto de endereços de e-mail existentes na base de referência. Devido à incompletude de nossa base, não pudemos avaliar todos os conjuntos de endereços que as heurísticas identificaram nas listas de discussão e comparamos apenas os conjuntos que continham ao menos um endereço em comum com a base de referência. A Tabela 1 apresenta as proporções entre o número de endereços da base de referência e os endereços de e-mail encontrados na lista de discussão de cada projeto.

Para analisar os resultados de cada heurística, utilizamos duas medidas de reconhecimento de padrões e de recuperação de informações: Precisão e Sensibilidade. Elas são as mesmas medidas utilizadas por Kouters [13] e por Goeminne et al. [9] para avalição das heurísticas de identificação de autores.

Essas medidas possibilitam a avalição dos acertos e erros das heurísticas em relação às informações existentes no gabarito e sua representatividade em relação ao número total de endereços de e-mail encontrados em cada lista de discussão. A precisão é a proporção dos endereços de e-mail que foram corretamente identificados de acordo com nosso conjunto verdade. Por outro lado, a medida de sensibilidade avalia a proporção dos endereços que identificamos como pertencentes a um membro da comunidade e que realmente pertencem a esse membro, de acordo com o gabarito. Um índice baixo de precisão significa que a heurística está atribuindo endereços de e-mail para os membros da comunidade que não os pertencem. Um índice baixo de sensibilidade significa que a heurística está errando por omissão e está deixando de atribuir endereços que notoriamente pertencem a dado um membro da comunidade.

Para calcular essas medidas, utilizamos uma matriz de confusão que possibilita a avaliação da qualidade dos resultados das heurísticas.

Tabela 2 - Matriz de confusão

	Positivo (conhecido)	**Negativo (conhecido)**
Positivo (resultado da heurística)	Verdadeiro Positivo (vp)	Falso Positivo (fp)
Negativo (resultado da heurística)	Falso Negativo (fn)	Verdadeiro Negativo (vn)

Neste trabalho, conforme Tabela 2, temos:

- O número de verdadeiros positivos (vp): endereços que a heurística identificou como pertencendo a uma mesma pessoa e que estão de acordo com os endereços existentes na base de referência;
- O número de falsos negativos (fn): endereços de e-mail que a heurística não atribuiu corretamente à pessoa que o possui na base de referência;
- O número de verdadeiros negativos (vn): endereços que a heurística identificou corretamente como não pertencentes à mesma pessoa e que estão de acordo com a base de referência;
- O número de falsos positivos (fp): endereços de e-mail que a heurística atribuiu incorretamente a uma pessoa, sendo que este endereço pertence a outra pessoa na base de referência.

A partir da matriz de confusão, calculamos a Precisão e a Sensibilidade utilizando as seguintes equações:

$$Precisão = \frac{vp}{vp + fp} \qquad Sensibilidade = \frac{vp}{vp + fn}$$

Consideramos a precisão e a sensibilidade simultaneamente para determinar a qualidade dos resultados das heurísticas. Para isso, utilizamos a Medida F que proporciona uma média harmônica entre precisão e sensibilidade. Essa medida possibilita a análise de perdas e ganhos existentes nos resultados gerados pelas heurísticas. Para calcular a Medida F utilizamos a seguinte equação:

$$Medida\ F = 2 * \frac{Precisão * Sensibilidade}{Precisão + Sensibilidade}$$

Com essas medidas, realizamos duas avaliações dos resultados das heurísticas, a primeira utiliza os dados de todo o histórico das listas de discussão. A segunda utiliza este mesmo histórico fracionado em períodos de 1 ano. Utilizamos essa segunda avaliação para não favorecer a heurística de Oliva et al. [17] que faz uso da frequência que as pessoas utilizam o mesmo nome na configuração de seus clientes de e-mail.

4. RESULTADOS E DISCUSSÃO

Conforme descrito na seção anterior, avaliamos 3 heurísticas de identificação de autores em 16 projetos de software livre da Fundação Apache. Pode-se observar na Figura 3 que a heurística utilizada por Oliva et al. possui uma precisão maior em relação às demais quando consideramos todo o histórico das listas de discussão. Isso se dá pelo baixo índice de falsos positivos. Ela não utiliza nenhuma similaridade na identificação dos endereços, por isso é pouco afetada por endereços de e-mail que sejam muito semelhantes.

Ainda na Figura 3, podemos observar os resultados da precisão na avaliação utilizando períodos anuais das listas. Todas as heurísticas possuem bons resultados. As heurísticas ingênua e de Bird et al. possuem, em média, uma precisão melhor do que quando consideramos todo o período da lista de discussão. A fragmentação do histórico em períodos menores também reduz o ruído que afeta a métrica de similaridade destas heurísticas. Podemos observar ainda que a heurística de Oliva et al. não sofreu perda de precisão pela fragmentação do conjunto de dados em períodos menores.

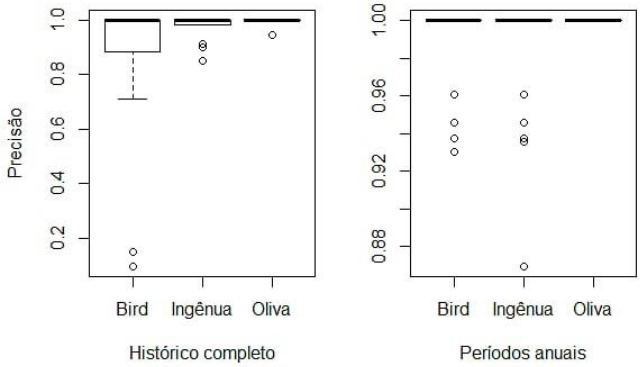

Figura 3 - Precisão das heurísticas considerando todo o histórico e os períodos anuais, respectivamente.

Por outro lado, conforme podemos observar na Figura 4, a heurística de Oliva et al. deixa de atribuir muitos endereços quando comparada com Bird et al. na avaliação que considera todo o histórico das listas de discussão. Apesar de a divisão do histórico em períodos anuais amenizar os falsos negativos, Oliva et al. ainda é menos eficaz na identificação de falsos negativos que Bird et al. Isso é devido, novamente, ao não uso de similaridade pela heurística de Oliva et al. O uso da similaridade possibilita que Bird et al. atribua corretamente endereços semelhantes, enquanto Oliva et al. não conseguem estabelecer o vínculo entre os endereços que não sejam idênticos.

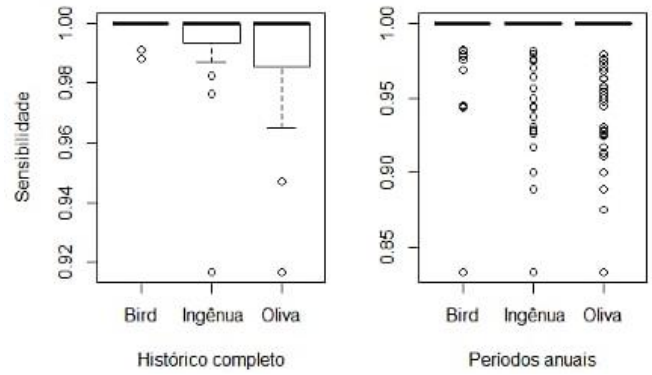

Figura 4 - Sensibilidade das heurísticas considerando todo o histórico das listas e por per períodos anuais, respectivamente.

A Medida F considera tanto os falsos positivos quanto os verdadeiros positivos. Podemos observar seus resultados de utilizando todo o histórico na Figura 5. Apesar de resultados modestos de sensibilidade, a heurística de Oliva et al. consegue uma menor variação na qualidade dos resultados quando consideramos todo o histórico na lista de discussão. Porém, novamente apresenta resultados inferiores quando fracionamos o histórico das listas anualmente.

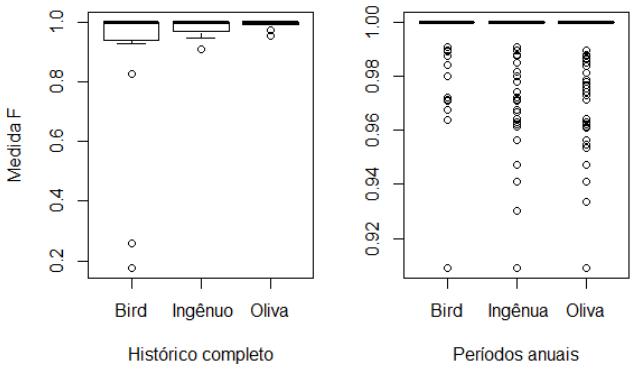

Figura 5 - Medida F das heurísticas considerando todo o histórico das listas e por per períodos anuais, respectivamente.

Existem indícios de que a heurística de Bird seja mais eficaz quando utilizamos um histórico de listas menor, contudo se estamos utilizando o histórico composto por vários anos, a heurística de Oliva et al. apresenta uma menor exposição a falhas. Além disso, podemos observar que as heurísticas possuem resultados piores quando consideramos um conjunto de dados maior. Para verificar a influência do tamanho da amostra nos resultados das heurísticas, utilizamos a correlação de Spearman entre os erros de cada heurística e o número de endereços de e-mail utilizados no conjunto de dados que a heurística utilizada para identificar os autores. Deste modo, avaliarmos a relação que os resultados das heurísticas sofrem pelo aumento do número de endereços de e-mail utilizados na comunidade.

Pode-se nas tabelas 3 e 4 que todas os erros das heurísticas apresentam uma relação entre média e alta quando consideramos todo o histórico das listas, mas que essa relação é baixa quando consideramos o histórico dividido em períodos. Esses resultados apresentam indícios de que os problemas de falsos positivos (fp) e falsos negativos (fn) podem ser amenizados com a utilização de um conjunto de dados menor no processo a identificação dos autores.

Tabela 3 - Correlação de Spearman entre erros das heurísticas e o número de endereços de e-mail envolvidos considerando todo o histórico das listas de discussão.

Heurística \ Medida	fp	fn
Bird	0,7508349	0,4886605
Ingênua	0,5081648	0,416205
Oliva	0,252236	0,5845726

Tabela 4 - Correlação de Spearman entre erros das heurísticas e o número de endereços de e-mail envolvidos considerando o histórico das listas de discussão divididos em períodos anuais.

Heurística \ Medida	fp	fn
Bird	0,22967	0,205483
Ingênua	0,213752	0,074241
Oliva	0	0,211868

Nossos resultados corroboram as afirmações de Bettenburg et al. [2] sobre a necessidade de inspeção manual para corrigir as falhas de identificação de autores pelas heurísticas na extração dados de listas de discussão. Contudo, ressaltamos que uma variação menor nas falhas facilita a inspeção manual por parte dos pesquisadores.

5. AMEAÇAS À VALIDADE

O objetivo deste trabalho foi comparar heurísticas considerando parte do ecossistema da Fundação Apache. A escolha dos projetos se deu pela disponibilidade de informações para criação da base de referência de cada projeto. Essa seleção pode incluir o viés da escolha dos projetos que foram utilizados na avaliação dos resultados dessas heurísticas. Contudo, foram incluídos projetos com diversidade de domínio, quantidade de pessoas e tempo de vida do projeto. Como trabalho futuro, realizaremos a avaliação das heurísticas com um conjunto maior de projetos para averiguar os resultados obtidos com essa amostra de 16 projetos.

A comparação considera apenas os endereços de e-mail que foram possíveis de serem adicionados à base de referência. Ela contém em média apenas 11% dos endereços de e-mail existentes na lista de discussão para cada projeto. A incompletude dessa base adiciona o viés sobre os falsos positivos identificados por cada heurística. Como trabalho futuro, ampliaremos nossa base de referência para conter uma proporção maior dos endereços de e-mail utilizados na lista e verificar os resultados obtidos pela nossa amostra atual do conjunto verdade.

Este trabalho considera os nomes de usuários do domínio apache.org como sendo equivalentes aos usuários do repositório de códigos fontes da Fundação Apache. Essa decisão se baseia na documentação existente na página da comunidade que afirma que "toda identificação única de submissão está vinculada a um endereço de e-mail da própria comunidade"[3] e na experiência adquirida durante a verificação manual dos nomes de usuários do repositório para o projeto Apache Ant.

A heurística de Oliva et al. se beneficia do uso de um volume maior de mensagens, enquanto que as demais sofrem com tal volume. Por esse motivo, realizamos uma das análises fracionando o conjunto de dados para amenizar o viés gerado pelo tamanho do histórico da lista de discussão. Como trabalho futuro, avaliaremos conjuntos menores de dados para estudar os comportamentos das heurísticas em períodos menores que um ano.

Se por um lado a diversidade de projetos favorece a generalização dos resultados, por outro pode servir de viés pelas diferentes características de cada comunidade. Devido à diversidade de número de usuários e tempo de vida de cada projeto, as heurísticas podem ser beneficiadas ou prejudicadas por estas características próprias de cada comunidade. Como trabalho futuro, avaliaremos o comportamento de cada heurística mediante a segregação das comunidades em conjuntos de listas de discussão com características semelhantes para mitigar o viés que essa diversidade pode incluir.

6. CONCLUSÕES

As heurísticas de desambiguação de autores em listas de discussão encontradas na literatura, em geral, utilizam apenas as informações das listas porque existem ocasiões em que informações adicionais sobre os membros não estão disponíveis. Nesse sentido, essas heurísticas podem ser utilizadas para identificação dos autores das mensagens em qualquer lista de discussão.

Utilizamos informações adicionais extraídas de diferentes locais (Jira, sites e repositório de chaves públicas da comunidade) para construir uma base de referência dos endereços de e-mail da comunidade e avaliar a capacidade de identificação de três dessas heurísticas: Bird et al.; Oliva et al. e a heurística ingênua considerada por Kouters em seus trabalhos. Realizamos duas análises para cada uma dessas heurísticas: a primeira utilizando todo o histórico da lista de discussão e a segunda fragmentando este histórico em períodos anuais. A divisão em períodos da segunda análise objetiva reduzir a vantagem que a heurística de Oliva et al. possui ao utilizar todo o histórico da lista de discussão. Lembramos que após a identificação automática realizada pelas heurísticas, é necessária a inspeção manual dos resultados para mitigar possíveis erros de atribuição. Contudo, quanto menor for o índice de erros da heurística utilizada, menor serão o esforço e tempo necessários para correção de possíveis falhas geradas na identificação de autores.

Encontramos indícios de que não existe uma única heurística de identificação de autores que seja melhor que as demais. O desempenho de cada heurística varia de acordo com o tamanho do conjunto de dados utilizado. Quando expomos as heurísticas a todo histórico de mensagens, Oliva et al. obtiveram melhores resultados de identificação dos autores. Contudo, quando dividimos o histórico em segmentos menores, a heurística de Bird et al. apresenta melhores resultados em comparação às demais.

A heurística de Bird et al. utiliza similaridade de nomes e endereços de e-mail para identificar os autores, enquanto que Oliva et al. evitam tal uso. Por este motivo, a primeira é mais bem-sucedida em um conjunto menor de dados que gerem menos ruídos durante a avaliação de similaridade dos autores. A segunda alcança melhores resultados em conjuntos de dados maiores porque é menos afetada pelo ruído existente.

Nossos resultados corroboram com os resultados de Goeminne et al. [9] que identificaram que um aumento nos endereços corretamente identificados implica no aumento de falsos positivos durante a identificação dos autores. Contudo, encontramos indícios de que o problema dos falsos positivos pode ser amenizado com a utilização de um conjunto de dados menor no processo a identificação dos autores.

Como trabalhos futuros, iremos reavaliar essas heurísticas comparando com as demais encontradas na literatura utilizando um número maior de projetos. Além disso, iremos explorar o modo como as características de cada comunidade (como tamanho e tempo de vida) podem influenciar na capacidade de identificação dessas heurísticas. Queremos com isso, facilitar ainda mais a escolha das heurísticas utilizadas pelos pesquisadores que utilizam as listas de discussão na condução de estudos científicos. Esse tipo de trabalho se torna cada vez mais relevante na medida em que aparecem numerosas comunidades de produção coletiva (e.g., MOOCs, projetos de software livre, comunidades de prática, etc.).

7. AGRADECIMENTOS

Agradecemos a Fundação Araucária, FAPESP, NAPSOL e NAWEB pelo apoio financeiro. Marco Gerosa recebe bolsa de pesquisador do CNPq. E agradecemos à Virgínia Carrara, estudante do Instituto de Matemática e Estatística da Universidade de São Paulo, pelo auxílio durante a análise dos resultados.

8. REFERÊNCIAS

[1] Bacchelli, A. et al. 2012. Content classification of development emails. *Software Engineering (ICSE), 2012 34th International Conference on* (2012), 375–385.

[2] Bettenburg, N. et al. 2009. An empirical study on the risks of using off-the-shelf techniques for processing mailing

[3] https://reference.apache.org/committer/email

list data. *Software Maintenance, 2009. ICSM 2009. IEEE International Conference on* (2009), 539–542.

[3] Bird, C. et al. 2008. Chapels in the bazaar? Latent social structure in OSS. *16th ACM SigSoft International Symposium on the Foundations of Software Engineering, Atlanta, GA* (2008).

[4] Bird, C. et al. 2006. Mining email social networks. *Proceedings of the 2006 international workshop on Mining software repositories* (2006), 137–143.

[5] Canfora, G. et al. 2011. Social interactions around cross-system bug fixings: the case of FreeBSD and OpenBSD. *Proceedings of the 8th working conference on mining software repositories* (2011), 143–152.

[6] D'Ambros, M. et al. 2008. Analysing Software Repositories to Understand Software Evolution. *Software Evolution*. T. Mens and S. Demeyer, eds. Springer. 37–67.

[7] Dendek, P.J. et al. 2013. Author disambiguation in the YADDA2 software platform. *Intelligent Tools for Building a Scientific Information Platform*. Springer. 131–143.

[8] Godby, C.J. et al. 2010. Who's who in your digital collection: developing a tool for name disambiguation and identity resolution. *Journal of the Chicago Colloquium on Digital Humanities and Computer Science* (2010).

[9] Goeminne, M. and Mens, T. 2013. A comparison of identity merge algorithms for software repositories. *Science of Computer Programming*. 78, 8 (2013), 971–986.

[10] Guzzi, A. et al. 2013. Communication in open source software development mailing lists. *Proceedings of the 10th Working Conference on Mining Software Repositories* (San Francisco, CA, USA, 2013), 277–286.

[11] Hassan, A.E. 2008. The road ahead for mining software repositories. *Frontiers of Software Maintenance, 2008. FoSM 2008.* (2008), 48–57.

[12] Hemmati, H. et al. 2013. The msr cookbook: Mining a decade of research. *Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on* (2013), 343–352.

[13] Kouters, E. 2013. Identity matching and geographical movement of open-source software mailing list participants. (2013).

[14] Kouters, E. et al. 2012. Who's who in Gnome: Using LSA to merge software repository identities. *Software Maintenance (ICSM), 2012 28th IEEE International Conference on* (2012), 592–595.

[15] Navarro, G. et al. 2001. Indexing methods for approximate string matching. *IEEE Data Eng. Bull.* 24, 4 (2001), 19–27.

[16] Nia, R. et al. 2010. Validity of network analyses in open source projects. *Mining Software Repositories (MSR), 2010 7th IEEE Working Conference on* (2010), 201–209.

[17] Oliva, G.A. et al. 2012. Characterizing key developers: a case study with apache ant. *Collaboration and Technology*. Springer. 97–112.

[18] Overbaugh, R.C. 2002. Undergraduate education majors' discourse on an electronic mailing list. *Journal of Research on Technology in Education*. 35, 1 (2002), 117–138.

[19] Panichella, S. et al. 2014. How Developers' Collaborations Identified from Different Sources Tell Us about Code Changes. *Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference on* (2014), 251–260.

[20] Pimentel, M. and Fuks, H. 2011. *Sistemas colaborativos*. Ed Campus.

[21] Rigby, P.C. et al. 2008. Open source software peer review practices: a case study of the apache server. *Proceedings of the 30th international conference on Software engineering* (2008), 541–550.

[22] Roberts, J. et al. 2006. Communication networks in an open source software project. *Open Source Systems*. Springer. 297–306.

[23] Robles, G. et al. 2009. Tools for the study of the usual data sources found in libre software projects. *International Journal of Open Source Software and Processes (IJOSSP)*. 1, 1 (2009), 24–45.

[24] Robles, G. and Gonzalez-Barahona, J.M. 2005. Developer identification methods for integrated data from various sources. *ACM SIGSOFT Software Engineering Notes*. 30, 4 (2005), 1–5.

[25] Squire, M. 2013. Project Roles in the Apache Software Foundation: A Dataset. *Proceedings of the 10th Working Conference on Mining Software Repositories* (San Francisco, CA, USA, 2013), 301–304.

[26] Steinmacher, I. et al. 2012. Newcomers withdrawal in open source software projects: Analysis of Hadoop Common project. *Collaborative Systems (SBSC), 2012 Brazilian Symposium on* (2012), 65–74.

[27] Xu, J. et al. 2005. A topological analysis of the open souce software development community. *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on* (2005), 198a–198a.

[28] Xuan, Q. and Filkov, V. 2014. Building it together: synchronous development in OSS. *Proceedings of the 36th International Conference on Software Engineering* (2014), 222–233.