UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE TECNOLOGIA

RICARDO ANTUNES BARBOSA

Análise de backlog de estórias de usuário por meio de agrupamento de textos e similaridade semântica

LIMEIRA

2016

RICARDO ANTUNES BARBOSA

Análise de backlog de estórias de usuário por meio de agrupamento de textos e similaridade semântica

Dissertação apresentada à Faculdade de Tecnologia da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Tecnologia na Área de Sistemas de Informação e Comunicação.

Supervisor/Orientador: Profa. Dra. Ana Estela Antunes da Silva

Co-supervisor/Coorientador: Profa. Dra. Regina Lúcia de Oliveira Moraes

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL
DA DISSERTAÇÃO DEFENDIDA PELO ALUNO
RICARDO ANTUNES BARBOSA, E ORIENTADA
PELA PROFA. DRA. ANA ESTELA ANTUNES DA SILVA

LIMEIRA
2016

Agência(s) de fomento e nº(s) de processo(s): Não se aplica.

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Faculdade de Tecnologia
Felipe de Souza Bueno - CRB 8/8577

Barbosa, Ricardo Antunes, 1982-
B234a Análise de *backlog* de estórias de usuário por meio de agrupamento de textos e similaridade semântica / Ricardo Antunes Barbosa. – Limeira, SP : [s.n.], 2016.

Orientador: Ana Estela Antunes da Silva.
Coorientador: Regina Lúcia de Oliveira Moraes.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Tecnologia.

1. Scrum (Desenvolvimento de software). 2. Desenvolvimento ágil de software. 3. Mineração de dados. 4. Engenharia de software. 5. Computação semântica. 6. Análise por agrupamento. I. Silva, Ana Estela Antunes da,1965-. II. Moraes, Regina Lúcia de Oliveira,1956-. III. Universidade Estadual de Campinas. Faculdade de Tecnologia. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: User stories backlog analysis through text clustering and semantic similarity
Palavras-chave em inglês:
Scrum (Computer software development)
Agile software development
Data mining
Software engineering
Semantic computing
Cluster analysis
Área de concentração: Sistemas de Informação e Comunicação
Titulação: Mestre em Tecnologia
Banca examinadora:
Ana Estela Antunes da Silva [Orientador]
Veronica Oliveira de Carvalho
Luiz Camolesi Júnior
Data de defesa: 15-12-2016
Programa de Pós-Graduação: Tecnologia

DISSERTAÇÃO DE MESTRADO EM TECNOLOGIA

ÁREA DE CONCENTRAÇÃO: SISTEMAS DE INFORMAÇÃO E COMUNICAÇÃO

Análise de *backlog* de estórias de usuário por meio de agrupamento de textos e similaridade semântica

Ricardo Antunes Barbosa

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:

_____

Profa. Dra. Ana Estela Antunes da Silva
FT/Unicamp
Presidente da Comissão Julgadora

_____

Profa. Dra. Veronica Oliveira de Carvalho
UNESP - Rio Claro

_____

Prof. Dr. Luiz Camolesi Júnior
FT/Unicamp

A ata da defesa com as respectivas assinaturas dos membros encontra-se no processo de vida acadêmica do aluno.

Ao meu Senhor Jesus Cristo que diariamente cuida de meus passos;

Aos meus pais, Joaquim e Ivete, que me ensinaram valores que hoje são a base de meu caráter;

A minha amada esposa Vanessa, que abriu mão de momentos em que poderíamos estar juntos para que eu pudesse desenvolver este trabalho;

Ao meu filho Vicente, fonte de inspiração, que, mesmo sem compreender, me presenteava diariamente com um sorriso encantador.

Agradecimentos

Esta dissertação não se faz apenas da redação de seu conteúdo e das ideias de seu autor. Nos bastidores haviam pessoas importantes que ajudaram a desenvolver este trabalho. Inicio apresentando os meus mais sinceros agradecimentos à minha família que, de diversas formas, contribuiu para este trabalho. Como? O simples ato de me dar ouvidos quando eu, empolgado, falava sobre este trabalho já era uma ajuda incrível!

Agradeço a Deus por me fazer lembrar das coisas que realmente importam na vida. Agradeço aos meus pais, Joaquim e Ivete, que, muitas vezes, me fizeram saber que estavam orgulhosos de mim e que, na simplicidade deles, se alegravam a cada novidade que eu lhes contava. Obrigado, pai, e Obrigado mãe, pelo incentivo e apoio incondicional.

Agradeço à minha orientadora Prof.ª Dr.ª Ana Estela Antunes da Silva pela confiança, incentivo e por pacientemente me conduzir na elaboração deste trabalho. Agradeço à coorientação da Prof.ª Dr.ª Regina Lúcia, cujas sugestões em pontos deste trabalho foram tão importantes. Agradeço aos professores participantes da minha banca de defesa, a Profa. Dra. Veronica Oliveira de Carvalho e ao Prof. Dr. Luiz Camolesi Júnior pelas importantes contribuições que guiaram a finalização deste trabalho. Estendo estes agradecimentos ao Prof. Dr. Paulo Sergio Martins Pedro, que junto com o Prof. Dr. Luiz Camolesi Júnior, participou de minha banca de qualificação e cujas sugestões colaboraram para a evolução deste trabalho. Sinto-me honrado em ter construído este trabalho junto a esses ilustres professores. Agradeço também aos professores e pesquisadores da Universidade de Coimbra com quem convivi e cujas sugestões contribuíram para o desenvolvimento deste trabalho. Um agradecimento especial à professora Cibele Oliveira que pacientemente me ajudou na revisão desta dissertação.

Agradeço aos amigos Daniel Vecchiato e Everaldo Leme pela ajuda dada para o desenvolvimento de alguns experimentos. A ajuda de vocês foi fundamental para a concretização deste trabalho. Sem falar dos momentos de descontração que tivemos em Coimbra que foram muito agradáveis.

E o que dizer a você, minha amada esposa Vanessa, que abriu mão de nossos passeios nos fim de semana e de outros muitos momentos de lazer para que meu desejo de fazer mestrado acadêmico se realizasse? Sem o teu carinho, compreensão e apoio, nada disso teria sido possível. Sem dúvida, o esforço para a concretização deste trabalho foi mais seu do que meu.

Vicente, meu filho. Obrigado pelos sorrisos que me alegraram em todos os dias que eu escrevia esta dissertação. Você é a maior bênção de minha vida!

“O conhecimento serve para encantar as pessoas, não para humilhá-las”
Mário Sérgio Cortella

Resumo

Scrum é uma metodologia ágil para gerenciamento e planejamento de projetos de software. O backlog do produto é um artefato do Scrum que consiste em uma lista priorizada de funcionalidades descritas geralmente na forma de estórias de usuário. A quantidade dessas estórias de usuário em um backlog está relacionada ao tamanho do sistema a ser desenvolvido e, em sistemas de larga escala, elas podem se apresentar em grande número. Mesmo as estórias de usuário sendo textos curtos, o elevado número de estórias dificulta a identificação de relacionamentos existentes entre elas e traz esforço adicional para: planejar sprints, para detectar casos de estórias duplicadas e para identificar estórias que representam alterações em uma estória prévia quando da inclusão de novos requisitos. Este trabalho teve por objetivo desenvolver e aplicar um processo para analisar estórias de usuário por meio da técnica de agrupamento de texto e análise de similaridade semântica. Os resultados indicam que o uso de agrupamento e uma métrica de similaridade semântica para análise de textos curtos é um método promissor para a descoberta de relacionamentos existentes entre estórias de usuário. Conclui-se que o uso desse processo contribui para: apoiar tarefas do Scrum como o refinamento do backlog do produto e o planejamento de sprints; apoiar a descoberta de estórias de usuário duplicadas e estórias que sofrerão evolução.

Palavras-chaves: *estórias de usuário, scrum, mineração de texto, agrupamento de texto, duplicidade, evolução, backlog do produto.*

Abstract

Scrum is an agile method for management and planning of software projects. The product backlog is a Scrum artifact, which consists of a prioritized list of features usually described in the form of user stories. The amount of these user stories in a product backlog is related to the size of the system to be developed and, in large-scale systems, they may be present in large numbers. Although user stories are represented by short texts, a large number of stories can make difficult the identification of relationships between them bringing additional effort: to plan sprints, to detect cases of duplicate stories and to identify stories which represent a change in a previous story when new requirements are included. This study aimed to develop and implement a process to analyze user stories through text clustering technique and semantic similarity analysis. The results indicate that the use of text clustering and the use of semantic similarity metrics for analysis of short texts is a promising method for the discovery of relationships between user stories. We conclude that the use of this process contributes to: support Scrum tasks as the refinement of the product backlog and plan the sprints; discover duplicate user stories and stories that undergo evolution.

Keywords: *user stories, scrum, text mining, text clustering, duplicity, evolution, product backlog.*

Lista de Figuras

Lista de Quadros

Lista de Tabelas

Lista de Siglas e Abreviaturas

ACM - *Association for Computing Machinery*

AGNES - *AGglomerative NESting*

CED - *Collins English Dictionary*

CRUD - *Create Read Update Delete*

COTUCA - Colégio Técnico de Campinas

DF - *Document Frequency*

ECI - Estórias em Cluster Independentes

ERP - *Enterprise Resource Planning*

IDF - *Inverse Document Frequency*

IEEE - *Institute of Electrical and Electronic Engineers*

KDT - *Knowledge Discovery from Text*

LDOC - *Longman Dictionary of Contemporary English*

LSO - *Lowest Super Ordinate*

MDSJ - *Multidimensional Scaling for Java*

MRD - *Machine Readable Dictionaries*

MSRP - *Microsoft Research Paraphrase Corpus*

OOPSLA - *Object Oriented Programming, Systems, Languages, and Applications*

PAM - *Partitioning Around Medoids*

PLN - Processamento de Linguagem Natural

ROI - *Return Of Investment*

RSC - Relacionamento *Sprint-Cluster*

SVM - *Support Vector Machines*

TF - *Term Frequency*

TF-IDF - *Term Frequency - Inverse Document Frequency*

VPML - *Visual Process Modeling Language*

VSM - *Vector Space Model*

Sumário

1 Introdução

Este capítulo apresenta alguns conceitos importantes para o entendimento deste trabalho, a motivação, os objetivos da pesquisa, assim como a organização desta dissertação.

1.1 Visão geral e motivação

O desenvolvimento ágil de *software* tem como objetivo acelerar a criação de um produto de *software* através de uma gestão mais leve com foco em entregas rápidas de incremento utilizável de produto. Os princípios do desenvolvimento ágil de *software* valorizam os indivíduos e suas interações, o funcionamento do *software*, a colaboração e o contato com o cliente e a rápida reação às mudanças de requisitos (Beck et al., 2001). As metodologias de desenvolvimento ágil produzem pouca documentação se comparadas com métodos tradicionais de desenvolvimento de *software*. Segundo Stettina e Heijstek (2011), a documentação mais reduzida se dá, pois as metodologias ágeis priorizam a comunicação face a face entre a equipe de desenvolvimento e o cliente.

Na metodologia ágil Scrum, as estórias de usuário são utilizadas para definir, gerenciar e documentar requisitos de usuário. As estórias de usuário são descrições de características ou funcionalidades do sistema a partir do ponto de vista do usuário (Cohn, 2004) e possuem informações sobre quem ou qual tipo de usuário está solicitando a funcionalidade, a funcionalidade que está sendo solicitada e a justificativa ou descrição do benefício dessa funcionalidade (Cohn, 2008). O *backlog* do produto é um artefato do Scrum que consiste em uma lista priorizada de funcionalidades a serem implementadas para um produto de *software* escritas na forma de estórias de usuário. É a partir do *backlog* do produto que serão extraídas as funcionalidades a serem implementadas dentro de um ciclo de trabalho do Scrum denominado *sprint*.

No início do projeto, o *backlog* do produto pode possuir poucos itens, mas, à medida que o *software* é desenvolvido e vai agregando valor, o *backlog* do produto recebe novos itens, sendo necessária sua manutenção e atualização. O *Backlog Grooming* ou *Reunião de Refinamento do Backlog* é uma reunião que tem como objetivo atualizar o *backlog* do produto, com intenção de mantê-lo preparado para o início do próximo *sprint* (Gregorio,

2012). Uma das atividades do *backlog grooming* é a descoberta de novas estórias e sua inserção no *backlog* do produto*,* e, quando necessário, a repriorização das estórias.

Já na fase de *planejamento do sprint* são reunidas as informações necessárias para estimar e entregar a implementação de um conjunto de itens priorizados do *backlog* do produto (Grapenthin et al., 2014). A quantidade de estórias selecionadas para o *sprint* é determinada pelo tamanho da funcionalidade e pela velocidade do time de desenvolvimento, e a seleção das estórias é feita seguindo a ordem de prioridade dos itens no *backlog* do produto. É durante o *planejamento do sprint* que se deve conhecer quais outras estórias possuem algum relacionamento com as estórias do *sprint,* para otimizar o desenvolvimento do produto e dar à equipe de desenvolvimento mais segurança de uma correta inclusão de uma estória a um dado *sprint.*

Um *software* complexo e de larga escala inclui grandes quantidades de código (Turk, France, & Rumpe, 2005) e geralmente possui muitos requisitos de usuário o que traz desafios para a análise, especificação e gerenciamento dos requisitos (Konrad & Gall, 2008). Assim, a especificação de requisitos seja ela qual for, como especificações escritas, por exemplo, pode representar um desafio a ser vencido, dados o volume e o esforço para o tratamento dos requisitos. Embora as estórias de usuário sejam uma proposta simplificada para captura de requisitos, um número elevado de estórias de usuário pode dificultar o entendimento, a seleção e a priorização dos requisitos. Nesse caso, o refinamento de um *backlog* do produto extenso pode ser uma tarefa complicada (Gregorio, 2012).

Apesar de algumas abordagens sugerirem a independência entre as estórias de usuário (Buglione & Abran, 2013; Leffingwell, 2011), dificilmente elas ficarão isentas de relacionamento entre si, principalmente em se tratando de grandes sistemas. Em cenários com muitas estórias de usuário, haverá dificuldades, por exemplo, na identificação de quais estórias de usuário estão relacionadas, seja em nível de complementação ou em nível de dependência.

O agrupamento de texto é uma tarefa da mineração de dados na qual um conjunto de textos é dividido em grupos de textos de conteúdo similares com o objetivo de se obter maior conhecimento sobre esses textos e sobre suas semelhanças (Guelpeli, 2012). A semelhança entre textos pode ser obtida com o uso de métricas na forma de funções de similaridade que retornam um escore indicativo do grau de similaridade existente entre dois

textos. Durante o agrupamento de requisitos textuais, uma função de similaridade é calculada como uma medida para a semelhança de requisitos (Güldali et al., 2011).

O uso de uma técnica de agrupamento de texto e de uma função de similaridade pode ajudar a analisar um grande conjunto de requisitos escritos na forma de estórias de usuário, para detectar características afins do sistema, duplicidades, alterações em requisitos, além de ajudar a descobrir informações úteis sobre o sistema a partir do relacionamento entre os requisitos.

1.2 Problemática

Em sistemas de larga escala desenvolvidos por uma metodologia ágil, a falta de conhecimento sobre as relações entre as estórias de usuário pode comprometer o planejamento dos *sprints,* pois a insuficiência desse conhecimento não garante à equipe de desenvolvimento a certeza de uma correta inclusão de uma estória em um *sprint*. A seleção de estórias para um *sprint* baseando-se apenas na ordem de prioridade pode não ser eficiente, já que a ordenação do *backlog* do produto pelo valor de negócio (do Inglês, *Business Value*) nem sempre considera os relacionamentos que possam existir entre as funcionalidades (Heidenberg et al., 2012). Nesse caso, podem existir itens no topo do *backlog* do produto que podem estar relacionados com itens de média ou baixa prioridade cuja inclusão no *sprint* não foi considerada. A identificação de estórias com características semelhantes facilitará a indicação de estórias a serem colocadas num mesmo *sprint* por meio da sugestão de possíveis trocas de estórias.

À medida que o produto de *software* é desenvolvido, novas funcionalidades são descobertas e inseridas no *backlog* do produto. Nesse momento, é importante saber se uma nova funcionalidade é semelhante a alguma já implementada ou a uma outra funcionalidade proposta. Em projetos de larga escala, estórias duplicadas podem surgir em razão do grande número de estórias geradas, da falta de comunicação entre os membros da equipe ou da própria velocidade de desenvolvimento imposta pelo Scrum. Nesse caso, uma das avaliações a ser feita no momento da chegada de uma nova funcionalidade é certificar-se de que ela não se trata de uma duplicidade, para evitar redundâncias no desenvolvimento. Identificar requisitos duplicados é uma tarefa importante para evitar atribuir o mesmo requisito a

diferentes desenvolvedores ou dar duas soluções a um mesmo problema (Natt och Dag et. al., 2002).

Outra avaliação importante é acompanhar a evolução que possa existir entre as estórias de usuário. Segundo Banerjee (2011), a evolução dos requisitos é um fenômeno natural no ciclo de vida de desenvolvimento de sistemas de *software*. Em se tratando de estórias de usuário, é fundamental verificar se uma nova estória de usuário é uma continuação ou alteração de uma estória de usuário já implementada, uma vez que, como qualquer outra forma de especificação de requisitos, as estórias de usuário também estão sujeitas à adição, remoção e alteração de funcionalidades. Nesses casos, será necessário comparar as novas estórias de usuário com as já implementadas, para detectar um possível caso de evolução. De acordo com Fabbrini et al. (2007), controlar a evolução dos requisitos escritos em linguagem natural é muito importante porque, em cada passo do caminho evolutivo dos requisitos, pode-se introduzir alterações indesejadas ou mesmo subtrair informações importantes. No caso de estórias de usuário, a simples identificação de situações em que houve uma evolução de requisitos já indicará ao analista de requisitos os pontos do sistema que possivelmente serão modificados.

O alto volume de estórias de usuário em um sistema de larga escala combinado com o dinamismo e velocidade com que as estórias chegam exigirá um maior dispêndio de tempo nas tarefas de refinamento do *backlog* do produto e planejamento de *sprints*. Por essa razão, faz-se necessária uma ferramenta automatizada que identifique os relacionamentos entre estórias e que sugira possíveis casos de duplicidade e evolução, para, dessa forma, apoiar tais tarefas do Scrum.

Não foram encontrados, na literatura, trabalhos que tratassem do problema de duplicidade de estórias de usuário ou trabalhos cuja proposta fosse identificar estórias de usuário que pudessem sofrer alguma alteração por meio da chegada de novas estórias. Da mesma forma, não foram encontrados casos de uso relacionados a duplicidade e evolução de estórias de usuário. Também não foram encontrados, na literatura, trabalhos que realizassem o agrupamento automático de estórias de usuário para auxiliar a análise de *sprints* nem mesmo trabalhos que empregassem alguma medida de similaridade para realizar a comparação de estórias de usuários.

Em geral, na literatura, há trabalhos que realizam agrupamento de requisitos na forma de documentação tradicional e trabalhos que utilizam alguma medida de similaridade

para analisar requisitos. No que se refere a estórias de usuário, existem poucos trabalhos que exploram o relacionamento existente entre requisitos neste formato. Alguns desses trabalhos foram analisados e serão apresentados nesta dissertação.

1.3 Objetivos

O objetivo geral é criar uma abordagem para analisar estórias de usuário a partir da técnica de agrupamento de textos e função de similaridade e apoiar a tomada de decisões dos participantes do Scrum em atividades realizadas na reunião de refinamento do *backlog* do produto e na reunião de planejamento de *sprints*.

Os principais objetivos específicos são: identificar casos de duplicação e evolução estórias de usuário, por meio do uso de função de similaridade, e sugerir a composição de *sprints* de desenvolvimento por meio do agrupamento de estórias de usuário.

1.4 Metodologia

Para o desenvolvimento da ferramenta de apoio à decisão, foram comparadas funções para cálculo de similaridade de texto com o propósito de identificar a função de similaridade que melhor observa semelhanças entre estórias de usuário. Para isso, uma base de textos curtos e de conjuntos de estórias de usuário reais foram analisados por cada uma das funções de similaridade escolhidas para o estudo.

Para a identificação de casos de duplicidade e evolução de estórias de usuário, foram utilizadas bases reais de estórias de usuário. A partir dessas bases, foram gerados exemplos de duplicidade e evolução de estórias, os quais foram analisados por uma função de similaridade. Cada exemplo de duplicidade e evolução gerado foi comparado com todos os itens de sua respectiva base de estórias por meio de uma função de similaridade. A partir de um limiar de similaridade indicado pelo usuário, foram sugeridos os possíveis casos de duplicidade e de evolução de estórias de usuário.

A ferramenta de apoio à decisão utilizou um algoritmo de agrupamento particional em combinação com a função de similaridade semântica WuP (Wu & Palmer, 1994), para

gerar grupos de estórias de usuário similares. Os dados gerados a partir da formação desses grupos e de algumas propriedades das estórias foram utilizados para sugerir ajustes em um *sprint* informado pelo usuário. Além disso, o resultado do agrupamento de estórias de usuário deve ser utilizado para indicar como estão relacionadas as estórias de usuário em um *sprint* (Seção 8.5).

A ferramenta de apoio à decisão foi implementada na linguagem de programação Java (versão 8) e utiliza funções para realizar atividades típicas de Processamento de Linguagem Natural (PLN) e funções para cálculo de similaridade em textos curtos.

1.5 Organização do texto

Este texto está organizado em capítulos. O Capítulo 1, já visto, apresentou uma introdução para contextualizar este trabalho, alguns conceitos importantes para a compreensão da problemática na qual se trabalhou e dos objetivos deste trabalho. Os capítulos 2, 3 e 4 dedicam-se ao referencial teórico utilizado para embasar esta pesquisa. Eles incluem, respectivamente, a base conceitual sobre a metodologia ágil Scrum; sobre a mineração de textos; e sobre a similaridade semântica. O Capítulo 5 reserva-se a apresentar os trabalhos relacionados a esta pesquisa. A metodologia de pesquisa e dos experimentos realizados neste trabalho são vistos no Capítulo 6. Os resultados dos experimentos e a discussão serão descritos no Capítulo 7. O Capítulo 8 dedica-se a apresentar a ferramenta desenvolvida neste trabalho. As conclusões e os trabalhos futuros são tratados no Capítulo 9.

2 Metodologia Ágil Scrum

Durante anos, os produtos de *software* foram construídos utilizando modelos tradicionais de gerenciamento de projetos. Nesses modelos tradicionais, um produto de *software* era todo planejado e documentado antes de ter sua implementação iniciada e todo o processo de construção seguia uma série de etapas bem definidas. Os modelos tradicionais, por exemplo, os modelos cascata e o espiral são mais eficientes em cenários previsíveis em que se tem pleno conhecimento dos requisitos de *software* e da tecnologia a ser utilizada. Porém, quando se conhecem pouco os requisitos do *software* ou quando eles sofrem mudanças, temos um cenário dinâmico e, por vezes, complexo em que modelos tradicionais de gerenciamento de projetos podem apresentar certas limitações.

Nesse cenário de imprevisibilidade de requisitos, é necessário o uso de uma abordagem de desenvolvimento adaptativa e flexível que consiga responder rapidamente a mudanças que podem ocorrer nos requisitos. O Scrum é um *framework* simples que contém princípios e práticas que seguem uma abordagem adaptativa. O Scrum foi projetado para se adaptar às mudanças de requisitos durante o processo de desenvolvimento em intervalos curtos e regulares, priorizando as necessidades dos clientes (Sutherland & Schwaber, 2007).

Nesta seção, serão introduzidos os princípios da metodologia ágil. Também será apresentado o *framework* Scrum junto com seus personagens principais, as atividades básicas e seus artefatos mais utilizados.

2.1 Histórico

Desde o início da história do desenvolvimento de *software*, os produtos de *software* são construídos usando metodologias tradicionais, também conhecidas como pesadas ou orientadas à documentação. Ainda hoje, as metodologias tradicionais são utilizadas principalmente em cenários considerados estáveis e previsíveis, em que se é possível obter todo o conhecimento do sistema logo no início do projeto. No entanto, cenários como esses têm-se tornado cada vez mais raros e os sistemas modernos têm se tornado cada vez mais complexos, necessitando ser produzidos de forma mais rápida.

Nos dias atuais, a metodologia tradicional de desenvolvimento é frequentemente vista como ultrapassada e demasiadamente rígida. O funcionamento sequencial no formato de fases, em que cada fase é dependente da anterior torna a metodologia tradicional ineficiente para alguns projetos, uma vez que torna o processo de desenvolvimento lento e, por conseguinte, não atende às expectativas do cliente. Em projetos menores, com equipe reduzida, o trabalho com a documentação mostrou-se árduo e, por vezes, consumidor de mais tempo de trabalho que a própria implementação do sistema.

Diante dessa conjuntura, um grupo de dezessete profissionais que trabalhavam com metodologias chamadas "leves" reuniram-se em fevereiro de 2001 para propor uma alternativa ao desenvolvimento de *software* orientado a documentação. Durante a reunião, foram discutidos os fatores para o sucesso dos projetos de *software* com que cada profissional esteve envolvido, além das ações comuns tomadas por esses profissionais em seus projetos. As conclusões foram registradas em um documento popularmente conhecido como Manifesto Ágil (2015), que possui os conceitos fundamentais que guiam o desenvolvimento ágil de *software*.

O Manifesto Ágil foi redigido em 2001 na *Object Oriented Programming, Systems, Languages, and Applications* (OOPSLA), alguns anos depois do anúncio do Scrum (em 1996). É reconhecido, entre os especialistas, que o Scrum influenciou fortemente esse Manifesto (Pham & Van Pham, 2011).

2.2 Visão geral do Scrum

O termo Scrum tem sua primeira aparição no artigo publicado em 1986 por Hirotaka Takeuchi e Ikujiro Nonaka (Takeuchi & Nonaka, 1986). Os autores propuseram uma abordagem holística de trabalho em que equipes de um projeto deviam ser constituídas por pequenas equipes multifuncionais trabalhando juntas para um objetivo comum. Os autores compararam esta composição à formação scrum oriunda do jogo de rugby. Scrum é um conjunto de valores, princípios e práticas para ajudar uma equipe a desenvolver e entregar produtos em ciclos curtos, o que permite um *feedback* rápido, melhoria contínua e rápida adaptação a alterações (Scrum Alliance, 2016), desenvolvendo o trabalho de forma mais produtiva e com maior qualidade (Sutherland & Schwaber, 2007). Em outras palavras, Scrum

é uma metodologia de desenvolvimento ágil de *software* usada para gestão dinâmica de projetos complexos.

A metodologia ágil Scrum é baseada na abordagem "*lean*" (Womack, Jones, & Roos, 1990) para desenvolvimento de *software*, em que as equipes escolhem a quantidade de trabalho a ser feito e a melhor forma de fazê-lo. O Scrum foi formalizado por Ken Schwaber e Jeff Sutherland e é uma das metodologias ágeis de uso mais crescente sendo utilizado por empresas como Yahoo!, Microsoft, Google, Lockheed Martin, Motorola, SAP, Cisco, GE Medical, CapitalOne, etc. (Sutherland & Schwaber, 2007). Foi em 1993, na empresa Easel Corporation em que o Scrum foi utilizado pela primeira vez por times de desenvolvimento de *software* para a criação de uma ferramenta de análise e projeto orientado a objetos (Sutherland, 2004). O trabalho de Schwaber (Schwaber, 1997) foi um dos primeiros a definir formalmente o processo Scrum e suas fases, tornando o Scrum conhecido e adotado em diversos projetos de desenvolvimento de *software*.

Scrum é um processo iterativo e incremental estruturado na forma de ciclos de trabalho chamados *sprints* - uma espécie de *timebox* com o tamanho de 1 a 4 semanas, em que um conjunto de funcionalidades de um *software* é implementado. Em um projeto gerenciado com Scrum, as funcionalidades a serem implementadas são mantidas em uma lista chamada *Backlog* do Produto (do Inglês, *Product Backlog*). A cada início de um *sprint*, há uma reunião denominada *Planejamento do Sprint* (do Inglês, *Sprint Planning Meeting*) em que são priorizados os itens do *backlog* do produto e selecionadas as funcionalidades a serem desenvolvidas no próximo *sprint*. Essas funcionalidades selecionadas para o *sprint* são transferidas do *backlog* do produto para o *backlog* do *sprint* (do Inglês, *Sprint Backlog*), uma lista de tarefas que o time de desenvolvimento se compromete a executar.

Em todos os dias de um *sprint,* ocorre uma reunião chamada Reunião Diária (do Inglês, *Daily Scrum*), com o objetivo de divulgar o conhecimento sobre o que foi feito no dia anterior, o que será feito hoje e quais são os impedimentos existentes. No fim de um *sprint,* as funcionalidades implementadas são apresentadas em uma reunião denominada Revisão do *Sprint* (do Inglês, *Sprint Review Meeting*). Em seguida, ocorre uma reunião chamada *Retrospectiva do Sprint* (do Inglês, *Sprint Retrospective*) para identificar o que funcionou bem e o que precisa ser melhorado.

A Figura 1 apresenta o *framework* Scrum com todo o processo descrito anteriormente.

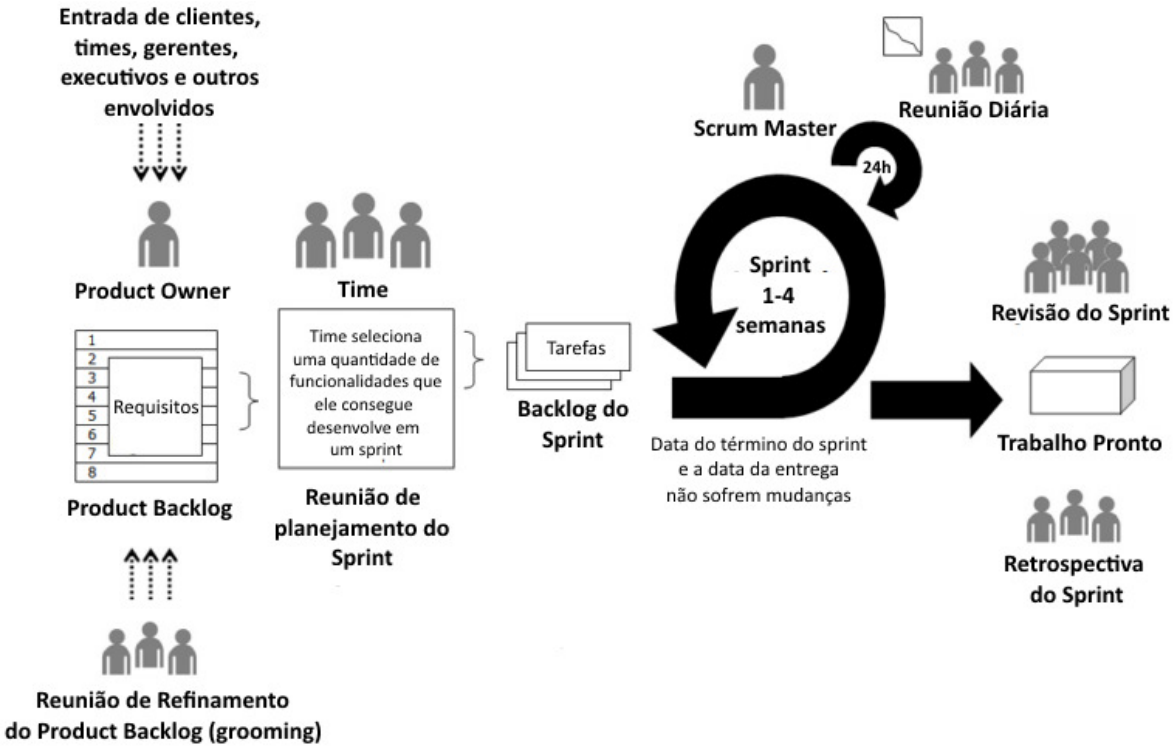


Figura 1 **-** *Framework* Scrum (traduzido e adaptado de Sutherland e Schwaber, 2007)

2.2.1 Papéis no Scrum

O time Scrum é composto por um Dono do Produto (do Inglês, *Product Owner*), por um Time de Desenvolvimento e por um *Scrum Master*. O time Scrum é auto-organizável, ou seja, o próprio time escolhe a melhor forma de realizar o trabalho ao invés de ser dirigido por alguém fora da equipe. O time Scrum é uma equipe multifuncional possuindo todas as competências necessárias para realizar o trabalho. O modelo de equipe no Scrum é projetado para otimizar a flexibilidade, a criatividade e a produtividade (Scrum Alliance, 2016).

O dono do produto é a pessoa responsável por definir as características do produto, decidir a sua data de entrega e o seu conteúdo. Ele é responsável por maximizar a rentabilidade do produto ou ROI (do Inglês, *Return Of Investment*) por meio da priorização

das funcionalidades que irão compor um sistema. O dono do produto é responsável por gerenciar o *backlog* do produto*,* o que inclui:

- Expressar claramente os itens do *backlog* do produto;
- Ordenar os itens do *backlog* do produto de forma a alcançar os objetivos;
- Otimizar o valor do trabalho que o time de desenvolvimento realiza;
- Garantir que o *backlog* do produto é visível, transparente e claro para todos; e
- Garantir ao time de desenvolvimento o entendimento dos itens do *backlog* do produto.

O dono do produto pode realizar mudanças nas características do sistema e rever priorizações a cada fim de *sprint*. É ele também que aceita ou rejeita o resultado do trabalho do time de desenvolvimento.

O time de desenvolvimento é composto por profissionais que trabalham para entregar um incremento utilizável do produto. No time de desenvolvimento, não existe necessariamente uma divisão por papéis como um arquiteto de *software*, programador, analista de sistemas, analista de testes, etc. O importante é que o time seja multidisciplinar, ou seja, os membros devem possuir todo o conhecimento necessário para poder entregar um incremento utilizável do produto. É o time que seleciona o *sprint* a ser desenvolvido e especifica os resultados do trabalho além de apresentar os resultados ao dono do produto. O time tem o direito de fazer tudo dentro do limite do projeto para alcançar o objetivo do *sprint*. O tamanho ideal para o time de desenvolvimento é pequeno o suficiente para permanecer ágil e grande o suficiente para completar um trabalho significativo dentro de um *Sprint* (Scrum Alliance, 2016). Para Sutherland e Schwaber (2007), o time de desenvolvimento deve possuir sete pessoas, podendo variar a composição com duas pessoas para mais ou para menos.

O *Scrum Master* é o facilitador do time de desenvolvimento. Ele é responsável por garantir que as regras e práticas do Scrum sejam seguidas pelo time de desenvolvimento. Segundo a Scrum Alliance (2016), as responsabilidades do *Scrum Master* frente ao dono do produto incluem:

- Encontrar técnicas para a gestão eficaz *backlog* do produto;
- Ajudar o time Scrum a compreender a necessidade de itens do *backlog* do produto de forma clara e concisa;
- Compreender o planejamento de produto em um ambiente empírico;

- Auxiliar o dono do produto a organizar o *backlog* do produto para maximizar o valor;
- Entender e praticar a agilidade; e
- Facilitar os eventos do Scrum conforme sejam solicitados ou necessários.

De acordo com a Scrum Alliance (2016), as responsabilidades do *Scrum Master* frente ao time de desenvolvimento incluem:

- Conduzir a equipe de desenvolvimento de modo auto-organizável e interfuncional;
- Ajudar a equipe de desenvolvimento a criar produtos de alto valor;
- Remover impedimentos para o progresso da equipe de desenvolvimento;
- Facilitar eventos Scrum conforme solicitado ou necessário; e
- Treinar a equipe de desenvolvimento em ambientes organizacionais em que Scrum ainda não seja totalmente adotado e compreendido.

Ainda segundo a Scrum Alliance (2016), as responsabilidades do *Scrum Master* frente ao time de desenvolvimento incluem:

- Liderar e treinar a organização na adoção Scrum;
- Planejar implementações Scrum dentro da organização;
- Ajudar os funcionários e partes interessadas a entender e aprovar Scrum;
- Efetuar mudanças que aumente a produtividade da equipe Scrum; e
- Trabalhar com outros *Scrum Masters* para aumentar a eficácia da aplicação do Scrum na organização.

2.2.2 Principais eventos e artefatos do Scrum

O Scrum possui quatro principais eventos: O Planejamento do *Sprint*, a Reunião Diária, a Revisão do *Sprint*, e a Retrospectiva do *Sprint*. A seguir, serão apresentados esses eventos.

No Planejamento do *Sprint* (do Inglês, *Sprint Planning*) são definidos quais itens do *backlog* do produto serão inseridos no próximo *sprint*. Dependendo do tamanho do projeto, o *backlog* do produto poderá ter um tamanho que represente algumas semanas ou até mesmo

meses de desenvolvimento. De acordo com Sutherland e Schwaber (2007), o planejamento do *sprint* se inicia com o dono do produto revendo a visão, o roteiro, o plano de lançamento, e o *backlog* do produto com a equipe Scrum. O time de desenvolvimento analisa as estimativas para as funcionalidades do *backlog* do produto e, em seguida, decide quanto trabalho é necessário para executar o *sprint* com base no tamanho da equipe, horas disponíveis, e o nível de produtividade da equipe. Após a seleção dos itens que comporão o *sprint*, o *Scrum Master*, junto ao time de desenvolvimento, transforma esses itens em tarefas e os insere no *backlog* do *sprint.*

É durante o planejamento do *sprint* que as funcionalidades de maior prioridade são selecionadas do *backlog* do produto para compor o próximo *sprint.* No entanto, a priorização do *backlog* do produto pelo valor de negócio nem sempre considera o relacionamento existente entre as funcionalidades. Diante disso, podem existir itens de alta prioridade (no topo do *backlog* do produto) que podem estar relacionados com itens de média ou baixa prioridade.

Durante todos os dias do *sprint*, os membros do time de desenvolvimento junto com o *Scrum Master* se reúnem para verificar como o *sprint* está progredindo. A Reunião Diária (do Inglês, *Daily Scrum* ou *Stand-Up Meeting*) é uma reunião geralmente realizada no mesmo horário com o tempo definido de aproximadamente 15 minutos. Nela o *Scrum Master* faz três perguntas para cada membro do time de desenvolvimento: "o que você fez ontem?", "o que fará hoje?" e "quais impedimentos estão em seu caminho?". O objetivo dessa reunião é ter uma visão geral do *sprint*, descobrir obstáculos, tratar necessidades individuais e fazer ajustes ao plano de trabalho.

No final de cada *sprint*, é realizada a Revisão do *Sprint* (do Inglês*, Sprint Review*). O objetivo desta reunião é verificar se é necessário realizar adaptações no produto. Durante essa reunião, o time Scrum e as partes interessadas discutem sobre o que foi feito no *sprint* e uma demonstração das novas funcionalidades é apresentada na forma de um incremento do produto. Em seguida, o dono doproduto determina quais itens do *sprint* foram concluídos e discute com o time Scrum qual a melhor forma de priorizar os itens do *backlog* do produto para o próximo *sprint*. O resultado da Revisão do *Sprint* é um *backlog* do produto revisado com definição dos prováveis itens para o próximo *sprint* (Scrum Alliance, 2016). O *backlog* do produto pode também ser ajustado para reconhecer novas oportunidades.

A reunião Retrospectiva do *Sprint* (do Inglês, *Sprint Retrospective*) ocorre depois da revisão do *sprint* e antes do próxima reunião de planejamento do *sprint*. O objetivo dessa reunião é verificar se é necessário realizar adaptações no processo de trabalho. Durante a retrospectiva do *sprint,* é verificado o que funcionou bem, o que pode ser melhorado e o que deve ser feito para melhorar. Segundo Scrum Alliance (2016), os propósitos da retrospectiva do *sprint* são:

- Inspecionar como foi o último *sprint* com relação a pessoas, relações, processos e ferramentas;
- Identificar e ordenar os itens principais que correram bem e as melhorias potenciais; e
- Criar um plano para implementação de melhorias para a forma de trabalho do time Scrum.

São três os principais artefatos utilizados no Scrum: *Backlog* do Produto, o *Backlog* do *Sprint* e o Gráfico de *Burndown*. A seguir, serão explicados esses três artefatos.

O *Backlog* do Produto é uma lista ordenada de todas as funcionalidades de um produto. O *backlog* do produto lista todas as características, funções, aprimoramentos e correções que constituem as mudanças a serem feitas no produto em versões futuras (Scrum Alliance, 2016). Geralmente o *backlog* do produto não está completo no início do projeto, possuindo apenas as funcionalidades mais evidentes. À medida que o produto é desenvolvido e se aprende mais sobre ele e sobre seus usuários, novas funcionalidades são adicionadas ao *backlog* do produto e repriorizações de itens são feitas. A descoberta de novas funcionalidades e a repriorização de itens são feitas durante a Reunião de Refinamento do *Backlog* do Produto (do Inglês, *Backlog Grooming* - livre tradução). Entretanto, antes das novas funcionalidades serem incluídas no *backlog do produto,* é necessário verificar se não se tratam de duplicações de funcionalidades já existentes no *backlog* do produto ou mesmo de correções ou complementações de alguma funcionalidade já desenvolvida.

O *backlog* do produto é um artefato em constante evolução, pois novos itens são adicionados e excluídos durante o desenvolvimento do produto. O dono do produto gerencia o *backlog* do produto, colocando os itens do *backlog* na ordem correta de implementação. Para realizar a priorização dos itens do *backlog* do produto, o dono do produto baseia-se em informações como valor ou importância da funcionalidade, custo, risco, tempo de implementação e retorno do investimento. A Figura 2 apresenta um exemplo simplificado de

um *backlog* do produto referente a um sistema de gerenciamento de empréstimos de livros de uma biblioteca.

Item	Descrição	Prioridade
1	Como aluno, eu quero reservar um livro pela internet para que eu não tenha que ir até a biblioteca	Alta
2	Como administrador, eu quero visualizar os livros que não foram entregues para que eu contate os alunos que os emprestaram	Alta
3	Como atendente, eu quero consultar a localização de um livro para que eu possa pegá-lo rapidamente	Media
4	Como atendente, eu quero consultar os dados dos alunos para poder contatá-los caso seja necessário	Média
5	Como administrador, eu quero solicitar empréstimo de livros entre bibliotecas para disponibilizá-los aos alunos	Média
6	Como aluno, eu quero colocar um comentário sobre o livro que li para que outros alunos possam ver	Baixa

Figura 2 - Exemplo simplificado de um *backlog* do produto

O *Backlog* do *Sprint* é uma lista de tarefas que o time de desenvolvimento compromete-se em realizar em um *sprint*. Os itens do *backlog* do *sprint* são retirados do *backlog* do produto, baseando-se nas prioridades que foram definidas pelo dono do produto. Os itens do *backlog* do *sprint* mostram ao time de desenvolvimento todo o trabalho que precisa ser realizado para atender ao objetivo do *sprint*. O *Scrum Master* é o responsável por manter e atualizar o *backlog* do *sprint* com informações que vêm da reunião diária. Dessa forma, o trabalho restante do *sprint* é estimado e apresentado em um gráfico de *Burndown*. A Figura 3 apresenta um exemplo simplificado de um *backlog* do *sprint* referente a um sistema de gerenciamento de empréstimos de livros de uma biblioteca.

		Horas de trabalho restantes para cada dia			
Descrição	**Tarefa**	**D1**	**D2**	**D3**	**...**
Como aluno, eu quero reservar um livro pela internet para que eu não tenha que ir até a biblioteca	Projetar tela de pesquisa de livro	8	4	0	
	Codificar módulo de reserva	6	4	2	
	Criar módulo para envio de e-mail	12	8	6	
	Criar comprovante de reserva	4	4	0	
Como administrador, eu quero visualizar os livros que não foram entregues para que eu contate os alunos que os emprestaram	Projetar tela consulta	10	4	2	
	Projetar relatório para exibir dados	12	6	0	
	Códificar módulo de impressão	12	8	2	
	Criar módulo para envio de e-mail	6	8	0	
	Total	**70**	**46**	**12**	

Figura 3 - Exemplo simplificado de um *backlog* do *sprint*

Diariamente o *backlog* do *sprint* é atualizado e o trabalho restante estimado é calculado e traçado pelo *Scrum Master*, resultando em um gráfico de *Burndown.* O gráfico de *Burndown* é uma ferramenta utilizada pelo time Scrum para acompanhar diariamente o progresso de um *sprint* e para prever quando todo o trabalho de um *sprint* será concluído. O gráfico mostra a quantidade de trabalho restante em um *sprint* (eixo vertical) versus o tempo do *sprint* (eixo horizontal). A partir do gráfico de *Burndown,* é possível verificar o quanto a equipe Scrum está adiantada ou atrasada com relação às estimativas feitas e entender a rapidez com que o time de desenvolvimento conclui as tarefas. A Figura 4 apresenta um exemplo de um gráfico *Burndown* para um *sprint* com duração de 20 dias.

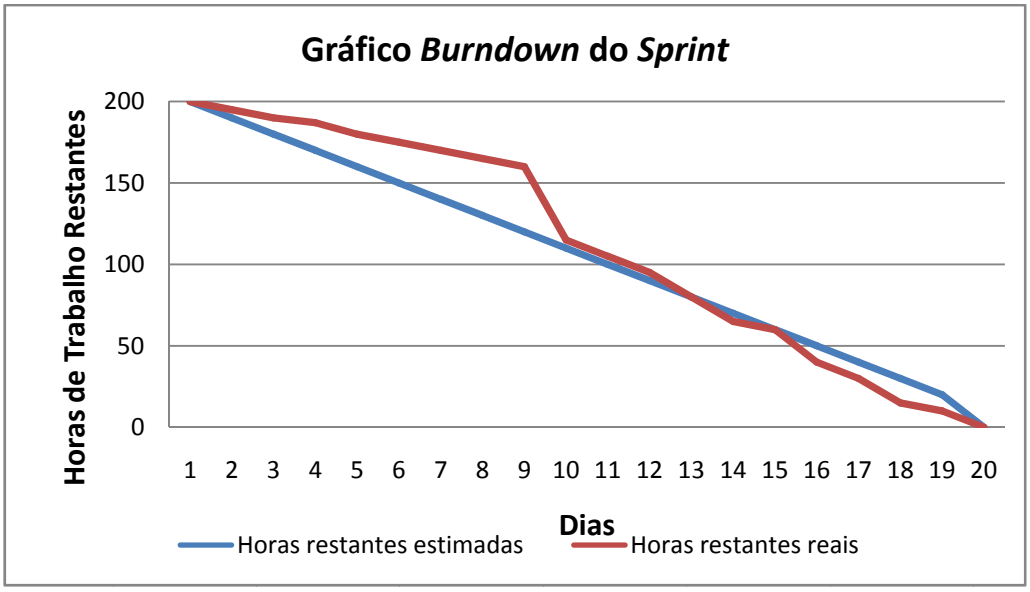


Figura 4 - Exemplo de gráfico *Burndown*

O gráfico representado na Figura 4 mostra diariamente o progresso do *sprint*. A linha diagonal azul traçada do ponto máximo até o ponto mínimo zero do eixo vertical representa as horas restantes estimadas. A linha vermelha representa as horas restantes reais. Percebe-se que, entre os dias 1 e 13, a linha vermelha permanece acima da linha azul, o que significa que nesse período houve um atraso em relação a estimativa. O período em que a linha vermelha está abaixo da linha azul (dias 16 a 19) houve uma antecipação em relação à estimativa. O gráfico mostra que, do dia 9 ao dia 10, houve uma redução acentuada das horas de trabalhos restantes. Essa recuperação pode acontecer quando uma força-tarefa conclui várias tarefas rapidamente ou quando, em uma negociação com o dono do produto, concorda-se em retirar algum requisito do *sprint*.

As funcionalidades de um *backlog* do produto costumam ser escritas na forma de estórias de usuários. A seção a seguir fará uma introdução sobre estórias de usuário e sobre os benefícios de seu uso.

2.3 Estórias de Usuário

As estórias de usuário são usualmente utilizadas nas metodologias ágeis para definir e gerenciar requisitos de usuário. As estórias de usuário são descrições de características ou funcionalidades do sistema a partir do ponto de vista do usuário (Cohn, 2004) sendo uma forma simplificada para registro da necessidade do usuário com relação a um produto. Uma estória de usuário geralmente possui informações sobre quem ou qual tipo de usuário está solicitando a funcionalidade, a funcionalidade que está sendo solicitada e uma justificativa ou descrição do benefício dessa funcionalidade (Cohn, 2008).

Em seu livro, Mike Cohn (2004) sugere um formato para a escrita de uma estória de usuário que é: "Como um <tipo do usuário>, eu quero <funcionalidade> para que <razão>", por exemplo, "Como um atendente *help desk*, eu quero pesquisar meus clientes pelo primeiro e último nome para que o tempo de atendimento ao cliente seja o menor possível". As estórias de usuário geralmente são escritas em cartões ou em papéis de anotação (*post-it notes*) e são colocadas em paredes ou quadros para facilitar o planejamento e a discussão sobre os requisitos. Em ferramentas de gerenciamento de projetos baseadas em Scrum, as estórias de usuário são registradas eletronicamente.

Estórias de usuário não fazem parte do *framework* Scrum. No Scrum, não há definição de um padrão para a escrita de requisitos e, portanto, o uso de estórias de usuário nesse *framework* é opcional. Apesar de terem sido originadas na metodologia de desevolvimento ágil XP (do Inglês, *Extreme Programming*), as estórias de usuário têm sido muito utilizadas no Scrum para descrever os itens do *backlog* do produto. Uma das razões para a adoção de estórias de usuário no Scrum é que elas são facilmente utilizadas para o planejamento do projeto, uma vez que, para cada estória é feita uma estimativa de quanto tempo e esforço são necessários para desenvolvê-la. Dessa forma, as estórias de usuário fracionam o problema, o que facilita a estimativa de esforço e tempo total de desenvolvimento.

O uso de estórias de usuário em metodologias ágeis enfatiza a comunicação verbal e encoraja o time de desenvolvimento a coletar detalhes sobre funcionalidade, o que intensifica o aprendizado sobre o sistema a ser desenvolvido. Por elas representarem pequenos pedaços de valor de negócio, as estórias de usuário permitem que o projeto total seja dividido em pequenos incrementos de produto. Por possuírem uma estrutura simples, as estórias de usuário promovem um maior envolvimento do cliente, já que são preferencialmente curtas e não exigem habilidades ou conhecimentos específicos para a sua escrita e manutenção.

Apesar das estórias de usuário apresentarem muitos benefícios, elas também apresentam algumas limitações. Por serem escritas em linguagem natural, elas podem ser imprecisas, o que pode gerar diferentes interpretações sobre um mesmo requisito. Outra limitação é que as estórias de usuário não descrevem detalhes sobre a interação do usuário com o sistema.

Segundo Buglione e Abran (2013) e Cohn (2004), boas estórias de usuário devem possuir os critérios INVEST propostos por William Wake (Wake, 2003). INVEST é o acrônimo de *Independent - Negotiable -Valuable - Estimable - Small -Testable* e representa as características que boas estórias de usuário devem ter. Segundo Wake (2003), as estórias de usuário devem ser independentes (*Independent*) sempre que possível, permitindo que elas sejam implementadas em qualquer ordem. As estórias de usuário devem ser flexíveis e negociáveis (*Negotiable*), ou seja, os seus detalhes devem ser discutidos sempre que possível, com intenção de atender às necessidades do cliente. As estórias de usuário devem sempre prover valor (*Valuable*) para o cliente em termos de solução. Toda estória de usuário deve ter condições de ser estimada (*Estimable*), para que seja priorizada adequadamente. As estórias

de usuário devem ser pequenas (*Small*) para facilitar as estimativas e permitir que sejam independentes. Além disso, as estórias precisam ser testáveis (*Testable*), para que seja possível confirmar o seu correto funcionamento.

Apesar de ser uma definição de alto nível de um requisito, a estória de usuário é a principal ferramenta utilizada como ponto de partida para o entendimento do sistema e para a tomada de decisões.

Este capítulo apresentou a metodologia ágil Scrum e como as estórias de usuários são utilizadas neste *framework*. Apesar do Scrum resolver problemas comuns às metodologias tradicionais, o Scrum por si só, não está imune aos problemas típicos do uso de requisitos textuais, como por exemplo, a duplicidade, a ambiguidade e a incompletude. A identificação de estórias duplicadas e estórias que evoluirão com a chegada de novas estórias passam a ser atividades essenciais quando o Scrum é empregado no desenvolvimento de sistemas de larga escala.

3 Mineração de Textos

A mineração de texto é um processo inspirado na mineração de dados com o objetivo de extrair, automaticamente, informações úteis em bases textuais. A principal contribuição da mineração de texto é a descoberta de informação relevante em textos. Neste capítulo, serão apresentados alguns conceitos e definições importantes sobre mineração de textos, as etapas do processo de mineração e suas principais tarefas. No final do capítulo, serão discutidos alguns problemas da engenharia de requisitos de *software* que podem ser apoiados por técnicas de mineração de textos.

3.1 Definição e tipos de análise de textos

A mineração de textos, também conhecida por Descoberta de Conhecimento em Textos (do Inglês, *Knowledge Discovery from Text*), foi apresentada, pela primeira vez, por Feldman e Dagan (Feldman & Dagan, 1995) e consiste em um processo apoiado por computador para a identificação de padrões a partir de textos não estruturados escritos em linguagem natural. Feldman e Sanger (2007) definem a mineração de textos como um processo de conhecimento intensivo em que um usuário interage com uma coleção de documentos ao longo do tempo, utilizando um conjunto de ferramentas de análise. Para ambas as definições, o principal objetivo da mineração de texto é descobrir informações úteis que possam estar implicitamente contidas em textos.

A mineração de textos utiliza algoritmos computacionais para identificar padrões ou anomalias implícitas em dados textuais que não poderiam ter sido descobertos com métodos tradicionais de pesquisa. Segundo Rezende et al. (2011), a mineração de textos permite a transformação desse grande volume de dados textuais em conhecimento útil, muitas vezes inovador para as organizações.

Em mineração de textos, os textos podem ser analisados semântica ou estatisticamente. A análise semântica considera o significado dos termos e o papel que eles possuem em uma sentença. Na análise semântica, a sequência em que os termos aparecem em uma sentença é verificada, a fim de identificar a função de cada termo em uma oração. A análise semântica faz uso de técnicas baseadas em processamento de linguagem natural. Segundo Ebecken et al. (2003), o emprego dessas técnicas justifica-se pela melhoria em

qualidade da mineração de textos; isso quando incrementado de um processamento linguístico mais complexo.

A análise estatística de textos baseia-se na frequência em que os termos aparecem no texto e é uma abordagem de análise que independe do idioma no qual o texto foi escrito. Nesse tipo de análise, os termos são valorados não levando em consideração a parte da sentença em que o termo está ou qual termo o precede. Essa forma de análise permite a utilização de modelos para representação de documentos que podem ser explorados por métodos de aprendizado estatístico dos dados.

3.2 Tarefas da Mineração de Textos

As principais tarefas de mineração de textos são: associação, classificação e agrupamento.

De acordo com Baghela e Tripathi (2012), a tarefa de associação tem por objetivo descobrir coleções de atributos de dados que estão estatisticamente associados aos dados subjacentes. Regras de associação visam extrair correlações do tipo "se A então B", em que A e B são conjunções distintas de pares atributo-valor. De acordo com Mahgoub (2006), regras de associação no contexto de mineração de texto podem ser construídas com base em palavras-chaves e coleção de índices de documentos. Segundo Mahgoub (2006), dado um conjunto de palavras-chaves $A=\{w_1, w_2, w_n\}$ e uma coleção de documentos $D=\{d_1, d_2,...d_n\}$ sendo cada documento d_i um conjunto de palavras-chaves tal que $d_i \subseteq A$, um documento d_i contém w_i se e somente se $w_i \subseteq d_i$. Uma regra de associação é uma implicação na forma $w_i \Rightarrow w_j$ sendo que $w_i \subset A$, $w_j \subset A$ e $w_i \cap w_j = \varnothing$. Para avaliar as regras de associação, duas medidas são consideradas: o suporte e confiança. Dada a regra $w_i \Rightarrow w_j$, o suporte representa o percentual de documentos na coleção D que contêm $w_i \cup w_j$, indicando a relevância da mesma. Já a confiança corresponde, dentre os documentos que possui w_i, a porcentagem de documentos que possui w_j, indicando a validade da regra.

A tarefa de classificação de textos visa atribuir documentos de textos a classes predefinidas. Por exemplo, em um sistema de notícias, cada nova notícia pode ser rotulada automaticamente em classes predefinidas, tais como notícia de esporte, finanças, política, entre outras. Hotho et al. (2005) sugerem alguns métodos para a realização de classificação de

textos: classificação por métodos estatísticos (p. ex. *Naïve Bayes*), classificação por similaridade (p. ex. *Nearest Neighbor Classification*), classificação por árvore de decisão (p. ex. *Decision Tree*) e classificação por máquinas de vetores de suporte. Para avaliar a classificação de textos, as medidas mais utilizadas são a precisão (*precision*), a revocação (*recall*) e a acurácia (*accuracy*). A precisão quantifica a porcentagem de documentos recuperados que são de fato relevantes, ou seja, que pertencem a uma classe específica. Já a revocação indica a fração dos documentos relevantes que são efetivamente recuperados.

A tarefa de agrupamento de textos consiste na identificação de grupos de documentos similares e é utilizada quando não se conhece previamente as categorias em que os textos se classificam. O objetivo é organizar um conjunto de documentos em grupos, baseando-se em uma medida de proximidade (Rezende et al., 2011). Os grupos de documentos formados são conhecidos como *clusters* e cada documento de um *cluster* deve ser mais próximo aos demais documentos do mesmo *cluster* do que a documentos dos outros *clusters* (Hotho *et al.*, 2005). A ideia principal é maximizar a similaridade intragrupos e minimizar a similaridade intergrupos. Existem basicamente duas estratégias para realização de agrupamento, que são o agrupamento hierárquico e o agrupamento particional. Maiores detalhes sobre as estratégias de agrupamento serão descritos na seção 3.3.2.

3.3 Etapas do Processo de Mineração de Textos

Segundo Rezende et al. (2011), o processo de mineração de textos para extração e organização não supervisionada de conhecimento pode ser dividido em três fases principais, conforme apresentado na Figura 5: Pré-Processamento dos Documentos, Extração de Padrões com Agrupamento de Textos e Avaliação do Conhecimento.

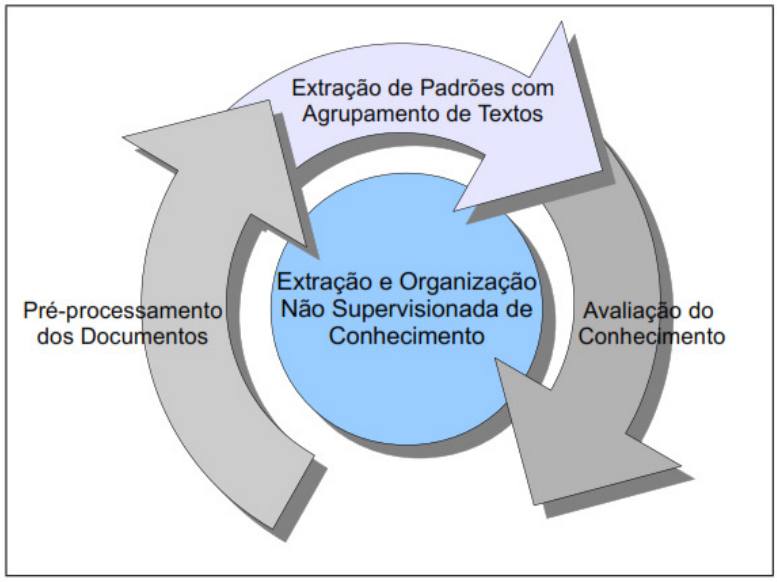


Figura 5 - Fases principais da Mineração de Textos (Rezende et al., 2011)

3.3.1 Fase de Pré-processamento dos Documentos

Os documentos que são submetidos ao processo de mineração de textos são textos não-estruturados. Na fase de Pré-processamento, os documentos são padronizados para se obter uma estrutura que possa representá-los e que permita extrair informações que ajudem a interpretá-los. O principal objetivo da fase de Pré-processamento dos Documentos é conseguir uma representação padronizada dos documentos que possa ser utilizada por algoritmos de agrupamento de textos. Para isso, são realizadas atividades de preparação do texto, seleção de termos principais e, finalmente, a representação estruturada dos textos.

Durante sua preparação, o texto é transformado em texto plano (sem formatação) e é organizado de forma que as palavras que o compõem possam ser extraídas adequadamente. Neste momento, o texto deve ser dividido em componentes significativos por meio do processo denominado Tokenização (*Tokenization*). A Tokenização é um processo frequentemente utilizado em sistemas de mineração de textos (Feldman & Sanger, 2007) e consiste em dividir o texto em *tokens,* como, por exemplo, palavras no idioma em que a extração está sendo realizada. *Token* é um exemplo de uma sequência de caracteres de um documento, que são agrupados em uma unidade semântica útil para processamento (Manning, Raghavan, & Schütze, 2008).

O processo de seleção de termos consiste em extrair um conjunto de palavras (termos) da coleção de documentos para compor o modelo de representação dos textos. Esse é um processo difícil, porque devem ser escolhidos os termos mais significativos para melhor representar os documentos, e é também um processo muito importante, pois, nos documentos, existem muitos termos irrelevantes que, se considerados, aumentariam o tempo de processamento na fase de extração de padrões. O processo de seleção de termos geralmente inicia-se com a remoção de palavras pouco representativas ou muito comuns que apresentam pouca relevância para a representação dos documentos. Tais palavras estão contidas em uma lista denominada *stopwords* e são desconsideradas no processo de seleção de termos. Esse processo de descarte de palavras comuns é conhecido como remoção de *stopwords*. Em seguida, os termos restantes passam por um processo de normalização, que pode ser a "lematização" (*lemmatization*) e/ou a "estemização" (*stemming)*.

O objetivo dos processos lematização e estemização é reduzir as palavras flexionadas ou derivadas à sua forma base comum, promovendo uma espécie de redução de dimensionalidade semântica. Segundo Manning et al. (2008), no processo de lematização é realizada uma análise morfológica da palavra para se recuperar seu lema, que é a forma na qual a palavra aparece no dicionário; ao passo que, no processo de estemização*,* se faz uso de uma heurística simples, que basicamente "corta" das palavras seus afixos de acordo com um conjunto de regras, e retorna sua forma raiz. Na lematização, por exemplo, a palavra em Inglês *best* tem como lema a palavra *good* enquanto que, na estemização*,* essa associação não é possível, já que, nesse processo, não se faz uso de um dicionário. Já a palavra em Inglês *talking* possui *talk* como palavra base, sendo que essa associação é conseguida tanto pela lematização quanto pela estemização.

O processo estemização remove dos termos partes como sufixos, plurais e gerúndios, deixando-os em sua forma base ou raiz. Por exemplo, no idioma Inglês, a palavra *count* pode aparecer em formas diferentes como *counting, counter, countable, counted*, etc., e, por possuírem semântica similares, serão reduzidas a sua forma raiz (*count*) após serem submetidas à estemização. A estemização colabora para reduzir o número de palavras a serem consideradas, uma vez que, se encontradas duas ou mais palavras com a mesma forma raiz, apenas a palavra na forma raiz será considerada para compor a representação do texto. Entre os algoritmos de estemização para a língua inglesa, destacam-se: o algoritmo *S Stemmer* (Harman, 1991)*,* o algoritmo de Lovins (Lovins, 1968) e o algoritmo Porter (Porter, 1980).

Embora os processos de remoção de *stopwords* e estemização reduzam o número de palavras a serem analisadas, esses processos podem não ser suficientes para garantir uma adequada seleção de termos e uma redução satisfatória da dimensionalidade do modelo de representação dos textos. Uma abordagem para reduzir o número de termos a serem utilizados em um modelo de espaço vetorial pode ser baseada na frequência absoluta de um termo em um conjunto de documentos. Nesta abordagem são contabilizadas as ocorrências de cada termo e, em seguida, os termos são classificados, permitindo que os *n* termos mais frequentes e os *n* termos menos frequentes sejam descartados. Os termos restantes, ou seja, os que não são muito e nem pouco frequentes serão considerados termos relevantes. Essa abordagem pode ser usada se assumirmos que termos com alta frequência não são importantes porque eles aparecem na maioria dos documentos. Da mesma forma, assume-se que os termos com baixa frequência não são significativos o suficiente por se tratarem de termos raros.

Outra abordagem para a seleção de termos pode basear-se na frequência do documento (do Inglês, *Document Frequency* - DF), que representa o número de documentos que contém um determinado termo. Nessa abordagem, serão selecionados os termos que aparecem em, no mínimo, *n* documentos da coleção. Nesse caso, define-se previamente um ponto de corte *n*, que indica o número mínimo de documentos em que o termo deve aparecer. Assim, serão descartados os termos cuja frequência de documentos for menor que um ponto de corte definido. Essa abordagem pode ser utilizada se assumirmos que termos que aparecem em poucos documentos não são informativos o bastante para representar os documentos da coleção.

Após a seleção dos termos, os documentos precisam ser representados de forma estruturada para que algoritmos computacionais possam analisá-los. Inicialmente proposto por Salton (Salton, Wong, & Yang, 1975), o Modelo de Espaço Vetorial (do Inglês, *Vector Space Model*) é uma forma de representar algebricamente documentos de texto por vetores de termos ponderados em um espaço *n*-dimensional. No modelo de espaço vetorial, cada documento é representado por vetores de termos em que cada termo armazena um valor que representa um peso para esse termo no documento. Na Figura 6, vemos no modelo de espaço vetorial, que os termos selecionados se comportam como eixos e os documentos como pontos nesse espaço vetorial.

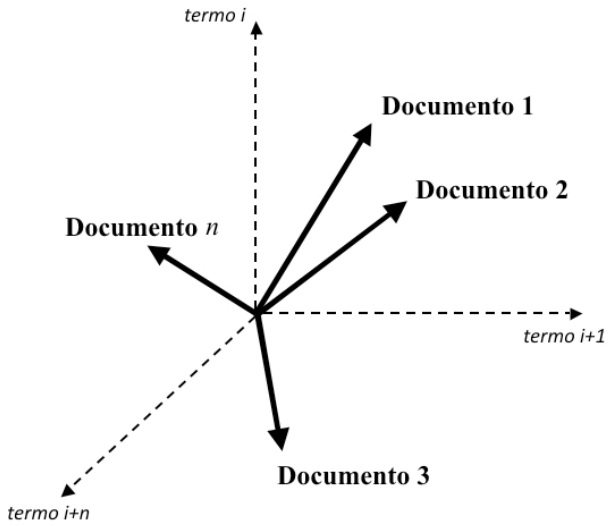


Figura 6 - Modelo de Espaço Vetorial

No modelo de espaço vetorial, os termos deverão estar associados a um valor (peso do termo) que indicará a sua importância dentro do documento em que aparece. O produto da frequência do termo pela frequência inversa do documento - mais conhecido como TF-IDF (do Inglês, *Term Frequency - Inverse Document Frequency*) - é uma das formas mais utilizadas para calcular e representar os pesos dos termos nos documentos dentro de um modelo de espaço vetorial.

O TF-IDF é uma medida estatística que diz o quão importante é um termo em um documento, considerando toda a coleção de documentos. O TF-IDF é diretamente proporcional à frequência do termo no documento em que aparece e inversamente proporcional à frequência do termo em toda a coleção de documentos. Os termos com números altos de TF-IDF indicam uma forte relação com os documentos em que aparecem (Ramos, Eden, & Edu, 2003).

O TF-IDF é composto por duas partes: Frequência do Termo (do Inglês, *Term Frequency* - TF), que calcula quantas vezes um termo ocorre em um documento específico, e Frequência Inversa do Documento (do Inglês, *Inverse Document Frequency* - IDF), que indica o quão importante é um termo considerando toda a coleção de documentos. Para evitar um viés para documentos mais longos - uma vez que um termo tem maiores chances de aparecer mais vezes em documentos extensos - o TF é normalizado dividindo-se a frequência

do termo pelo comprimento do documento. O TF normalizado pode ser definido conforme a Equação 1:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{k=1}^{K} n_{k,d}} \tag{1}$$

sendo $f_{t,d}$ o número de ocorrências do termo *t* em um documento *d* e *K* a quantidade de termos distintos no documento.

O IDF considera a raridade de um termo em uma coleção. Essa é uma medida que favorece termos que aparecem em alguns poucos documentos. Por exemplo, se um determinado termo aparece em todos os documentos da coleção, seu valor IDF é zero, enquanto que o valor IDF é alto para um termo raro. O IDF de um termo *t* pode ser definido conforme descrito na Equação 2:

$$idf(t) = log\frac{D}{df(t)} \tag{2}$$

sendo *D* o número total de documentos na coleção e *df*(*t*) o número de documentos que contém o termo.

O TF-IDF é o resultado do produto entre *tf* (parâmetro local) e *idf* (parâmetro global). Isso dá um peso para um termo *t* no documento *d* que é definido conforme Equação 3.

$$tfidf(t,d) = tf(t,d) \times idf(t) \tag{3}$$

Por exemplo, dado um documento *D* com 100 palavras em que o termo *computador* aparece 5 vezes, a frequência do termo (TF) para o termo *computador* é (5/100) = 0,05. Em uma coleção de 1.000 documentos em que o termo *computador* aparece

em 10 deles, a frequência de documento inversa (IDF) é calculada como *log*(1.000/10) = 2. Desta forma, o TF-IDF da palavra *computador* no documento *D* é (0,05 * 2) = 0,1.

O valor de TF-IDF será alto quando a frequência do termo é um valor alto e a frequência do documento na coleção é um valor baixo. A seção a seguir apresentará a fase de extração de padrões por meio da técnica de agrupamento de textos.

3.3.2 Fase de Extração de Padrões com Agrupamento de Textos

Uma das formas mais utilizadas para se extrair conhecimento de um conjunto de dados é por meio da verificação de semelhanças ou padrões entre eles. O agrupamento por semelhança, além de ser intuitivo ao ser humano, é um método preliminar ideal para análise de objetos (Embrechts, et al., 2013). Na fase de extração de padrões, pode-se utilizar um algoritmo computacional de agrupamento de textos para formar grupos de documentos textuais semelhantes.

O agrupamento de texto é uma tarefa da mineração de dados na qual um conjunto de textos é dividido em grupos de textos de conteúdo similar com o objetivo de se obter maior conhecimento sobre tais textos e sobre suas semelhanças (Guelpeli, 2012). No processo de agrupamento, os textos semelhantes são colocados em um mesmo conjunto, formando um grupo de documentos coesos, enquanto que documentos não relacionados a esse conjunto formarão outros grupos coesos. A ideia principal desse processo é maximizar a similaridade intragrupo e minimizar a similaridade intergrupo. O agrupamento de textos é uma abordagem que ajuda a entender como documentos textuais estão relacionados e ajuda a identificar o conhecimento implícito útil e possíveis conteúdos repetitivos.

No processo de agrupamento de texto, dois elementos importantes devem ser considerados: a medida de proximidade e a estratégia de agrupamento. A medida de proximidade diz respeito à forma como será calculada a distância ou similaridade entre textos, enquanto que a estratégia de agrupamento trata da forma como o texto será agrupado, o que influenciará na escolha do algoritmo de agrupamento. A seguir, serão apresentadas algumas medidas de proximidade, as estratégias de agrupamento e os algoritmos *k-means* e *k-medoids*.

Medidas de proximidade

As medidas de proximidade fornecem valores numéricos que indicam a distância entre dois objetos. Em problemas de agrupamento, por exemplo, quanto menor a distância entre os objetos, mais similares eles são, e, consequentemente, tendem a ficarem juntos. Por outro lado, quanto maior for o distanciamento entre os objetos, menos similares eles são, e, por essa razão, devem ser colocados em grupos distintos.

As medidas de proximidade podem calcular tanto a similaridade quanto a dissimilaridade entre objetos (Rezende et al., 2011). O cálculo de similaridade é um componente básico para muitas aplicações de mineração de texto (B. Li & Han, 2013). Em tarefas que envolvam análise de textos, as medidas de proximidade são, muitas vezes, adotadas como medidas ou funções de similaridade de textos. A seguir, serão apresentadas três medidas de proximidade/distância normalmente utilizadas em problemas de agrupamento e análise de textos: Distância Euclidiana, Similaridade pelo Cosseno do Ângulo (do Inglês, *Cosine Similarity*) e Coeficiente de Jaccard.

A distância euclidiana corresponde à distância geométrica entre dois objetos em um plano multidimensional. Dados os documentos *d1* e *d2,* representados pelos seus vetores $\overrightarrow{d1}$ e $\overrightarrow{d2}$ respectivamente, a distância euclidiana entre os documentos $\overrightarrow{d1}$ e $\overrightarrow{d2}$ num espaço *n*-dimensional é definida conforme descrito na Equação 4:

$$Dist(\overrightarrow{d1}, \overrightarrow{d2}) = \sqrt{\sum_{t=1}^{n} |w_{t,1} - w_{t,2}|^2} \quad (4)$$

sendo $w_{i,j}$ o peso do termo t_i no documento d_j e n o número de termos. O TF-IDF pode ser utilizado para calcular os pesos dos termos.

Embora a distância euclidiana seja amplamente utilizada em problemas de agrupamento, inclusive em agrupamento de texto (A. Huang, 2008; Song & Li, 2006), ela não é uma boa alternativa para verificar a similaridade entre documentos representados em um

modelo de espaço vetorial. A principal desvantagem da distância euclidiana no modelo de espaço vetorial é que essa medida é facilmente influenciada pelo tamanho dos documentos, ou seja, a distância euclidiana entre os vetores $\overrightarrow{d1}$ e $\overrightarrow{d2}$ pode ser elevada, mesmo quando a distribuição dos termos em $\overrightarrow{d1}$ e $\overrightarrow{d2}$ é semelhante. Por exemplo, dado um documento *d'* contendo o dobro do conteúdo do documento *d,* a distância euclidiana entre eles será alta, mesmo que esses documentos contenham grande número de palavras iguais.

A similaridade pelo cosseno do ângulo (*Cosine Similarity*) é uma das medidas mais utilizadas para quantificar a similaridade entre dois documentos em um modelo de espaço vetorial. Esta métrica é calculada por meio do cosseno do ângulo entre dois vetores. O seu cálculo é muito eficiente, especialmente para vetores esparsos, uma vez que apenas os valores diferentes de zero precisam ser considerados (B. Li & Han, 2013). Dados dois vetores *v* e *w*, ambos com mesma dimensão, o cosseno do ângulo formado por eles é obtido conforme Equação 5.

$$cosine(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{n} v_i \times w_i}{\sqrt{\sum_{i=1}^{n} {v_i}^2}\sqrt{\sum_{i=1}^{n} {w_i}^2}} \quad (5)$$

Como resultado, a similaridade por meio do cosseno do ângulo é um valor não negativo entre 0 e 1. Quando o ângulo formado entre o $\vec{v}$ e $\vec{w}$ está próximo de zero grau, o valor do cosseno desse ângulo tende a 1, ou seja, 100%, o que significará que os documentos representados por esses vetores são muito semelhantes. Por outro lado, quando se aumenta o ângulo formado pelos vetores, o valor do cosseno deste ângulo diminuirá, tendendo a zero (0% de similaridade), o que indica uma redução da similaridade. Na Figura 7, é possível verificar que o ângulo formado entre os documentos 1 e 2 é próximo de zero grau, o que indica que esses documentos são semelhantes. Verifica-se também que o ângulo formado entre os documentos 2 e 3 sugere que tais documentos possuem pouca similaridade.

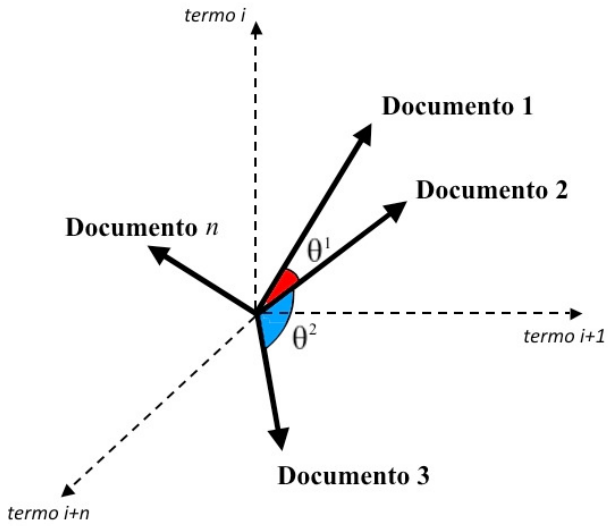


Figura 7 - Exemplo de documentos e ângulos formados em um modelo de espaço vetorial

A dissimilaridade com base no cosseno do ângulo pode ser definida conforme descrito na Equação 6.

$$dissim_{cos} = 1 - \ cosine(\vec{v}, \vec{w}) \tag{6}$$

Apesar da similaridade pelo cosseno do ângulo ser dependente do modelo de espaço vetorial, ela apresenta vantagens em relação a outras medidas de distância, como, por exemplo, a distância euclidiana. Segundo Huang (2008), a principal vantagem da função de similaridade pelo cosseno do ângulo em relação à distância euclidiana é que a primeira não é influenciada pelo comprimento do documento. Por exemplo, dado um documento *d',* contendo o dobro do conteúdo do documento *d*, a similaridade pelo cosseno do ângulo entre *d'* e *d* é 1, indicando que *d'* e *d* são documentos idênticos. No caso do uso da distância euclidiana, os documentos *d'* e *d* possuiriam um distância alta entre si.

O coeficiente de Jaccard, também conhecido como índice de Jaccard (Jaccard, 1901) é uma medida de similaridade utilizada para medir a proporção de termos comuns entre dois documentos. O coeficiente de Jaccard é definido pelo número de termos que são comuns aos documentos (intersecção), dividido pelo número total de termos encontrados nos documentos (união). A definição formal de coeficiente de Jaccard é expressa na Equação 7:

$$jaccard(d_i, d_j) = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

(7)

em que d_i e d_j são documentos; f_{11} é o número de termos comuns entre d_i e d_j; f_{01} é o número de termos ausentes em d_i, mas presente no d_j; e f_{10} é o número de termos presentes em d_i, mas ausente em d_j.

O resultado para o coeficiente de Jaccard varia entre 0 e 1. Se o resultado for 0, significa que os documentos são totalmente diferentes, enquanto que o resultado sendo 1 significa que os documentos são idênticos. A dissimilaridade com base no coeficiente de Jaccard pode ser definida conforme a Equação 8.

$$dissim_{jac} = 1 - jaccard(d_i, d_j)$$

(8)

Nesta seção foram apresentadas três medidas de proximidade relacionadas ao contexto deste trabalho. Na literatura, uma revisão com mais medidas de proximidade pode ser encontrada em Cha (2007); Gomaa e Fahmy (2013); Huang (2008).

Estratégias de agrupamento

Em geral, os métodos de agrupamento podem ser classificados em *hierárquicos* e *particionais*. O agrupamento hierárquico funciona de forma iterativa criando *clusters* maiores ou menores por meio da união ou divisão de *clusters,* enquanto que o agrupamento particional tenta decompor diretamente o conjunto de dados em *clusters* distintos (Casamayor *et al.*, 2012).

Segundo Filippone et al. (2008), os métodos de agrupamento hierárquico são capazes de encontrar estruturas que podem ainda ser divididas em subestruturas e assim sucessivamente, de forma recursiva. O resultado é uma estrutura hierárquica de grupos conhecida como dendrograma em que cada nó representa um grupo de documentos. O

agrupamento hierárquico ainda pode ser dividido em *aglomerativo* com base em uma abordagem *bottom-up,* e *divisivo* com base na abordagem *top-down*.

No agrupamento hierárquico aglomerativo, cada documento é considerado um *cluster* distinto e novos *clusters* são formados por meio da união de pares de *clusters* similares até existir apenas um único *cluster*. No agrupamento hierárquico divisivo, todos os documentos pertencem a um único *cluster* que será dividido formando *clusters* menores até restarem apenas grupos com apenas um único documento. Nos métodos hierárquicos, não é necessário especificar previamente um número de *clusters*.

A Figura 8 apresenta um dendograma em que é possível visualizar o sentido da construção dos agrupamentos aglomerativo e divisivo.

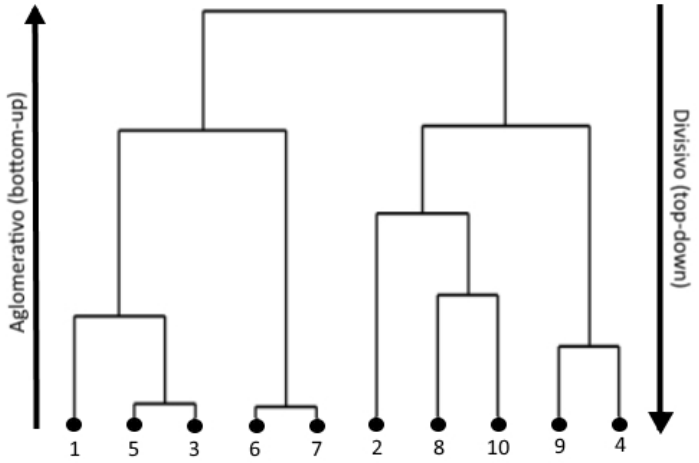


Figura 8 - Agrupamento hierárquico aglomerativo e divisivo

Os métodos de agrupamento particional procuram dividir um conjunto de objetos em *k clusters* onde *k* é especificado previamente. Os documentos são atribuídos a um *cluster* a partir de sua comparação com um ponto central do *cluster* chamado centroide. Esse ponto central é utilizado como parâmetro de comparação para que novos documentos se unam ao *cluster*. Nesse tipo de método, são utilizadas funções de similaridade que indicam o quanto um documento é semelhante aos documentos de um dado *cluster.*

O *k-means* (Macqueen, 1967) é o algoritmo de agrupamento particional mais conhecido e, a partir dele, foram geradas novas versões, otimizando o seu funcionamento,

como, por exemplo, o *k-medoids* (Kaufman & Rousseeuw, 1987), o *k-prototype* (Zhexue Huang, 1997), o *k-modes* (Z. Huang, 1998) e o *X-means* (Pelleg & Moore, 2000). Geralmente as variações do algoritmo original *k-means* modificam a forma como as médias iniciais são selecionadas, ou como é feito o cálculo de similaridade entre os objetos, ou o cálculo da média dos grupos.

Não se pretende nesta seção apresentar as variações do algoritmo *k-means* disponíveis na literatura ou mesmo os algoritmos de agrupamento hierárquico. Na literatura, uma revisão dos algoritmos de agrupamento hierárquico e particionais pode ser encontrada em Xu e Wunsch (2005).

Algoritmos *k-means* e *k-medoids*

O objetivo do agrupamento divisivo é determinar partições para *n* padrões em um espaço métrico *d*-dimensional dentro de *k* grupos, de modo que os padrões de um grupo são mais semelhantes entre si do que para padrões de outros grupos (Jain & Dubes, 1988).

Tanto o *k-means* (Macqueen, 1967) quanto o *k-medoids* (Kaufman & Rousseeuw, 1987) são algoritmos de agrupamento particionais que dividem um conjunto de objetos em *k* grupos, em que *k* é um valor predefinido. Ambos os algoritmos tentam minimizar a distância entre os objetos que serão colocados dentro de um *cluster* e um ponto designado como o centro do *cluster*. Os objetos que serão agrupados são atribuídos a um determinado *cluster* comparando-os com um ponto de referência dentro do *cluster*. No caso do *k-means* esse ponto de referência - conhecido como centroide - não é um objeto real do *cluster,* mas sim, uma média da distância entre todos os objetos do *cluster*. No caso do *k-medoids,* esse ponto de referência – também chamado de medoide - é um objeto real do *cluster,* que é escolhido a cada iteração do algoritmo para se tornar o elemento mais representativo do *cluster*. Tanto o *k-means* quanto o *k-medoids* precisam utilizar uma função de similaridade para indicar o quanto um objeto é similar ao ponto de referência do *cluster*.

O algoritmo *k-means* tem início com *k* centroides previamente selecionados e, a cada iteração, o *k-means* reconstrói o *cluster* (cada objeto é reposicionado no *cluster* com objetos mais similares) e recalcula os centroides até um ponto de convergência ser alcançado, ou seja, quando todos os objetos permanecem no mesmo *cluster*, mesmo depois da

atualização dos centroides. O funcionamento do algoritmo *k-means* pode ser visto no Quadro 1.

Quadro 1 - Algoritmo *k-means*

Algoritmo k-means

entrada: $O = \{o_1, o_2, \ldots, o_n\}$
k: número de *cluster*s

saída: $C = (c_1, c_2, \ldots, c_k)$

selecionar aleatoriamente *k* objetos para serem os centroides iniciais;
repetir
  para cada objeto $o \in O$ **faça**
    - calcular a (dis)similaridade entre *o* e cada centroide *C;*
    - atribuir *o* para o centroide mais próximo;
  fimpara
  recalcular os centroides para cada *cluster;*
até os *cluster*s não mudarem

Devido a forma como os centroides são calculados, o *k-means* é sensível a ruídos ou "*outliers*". Sendo um centroide a média de todos os objetos dentro de um *cluster*, se um objeto está muito distante dos demais, o centroide pode ser erroneamente influenciado.

Ao contrário do *k-means*, o *k-medoids* escolhe objetos reais para serem os pontos médios dos *cluster*s, no caso, medoides. Um medoide é um objeto real de um *cluster* cuja dissimilaridade é mínima a todos os outros objetos de um *cluster*. No *k-medoids,* mesmo se houver um objeto dentro de um *cluster* muito díspar, o custo de troca de medoide irá aumentar e o algoritmo irá descartar a troca do medoide atual pelo medoide recém calculado. Assim, o algoritmo *k-medoids* é menos sensível a *outliers* quando comparado com o algoritmo *k-means*.

A mais comum implementação do algoritmo *k-medoids* é a PAM (do Inglês *Partitioning Around Medoids*) (Kaufman & Rousseeuw, 1987). O funcionamento do algoritmo *k-medoids* pode ser visto no Quadro 2.

Quadro 2 - Algoritmo *k-medoids*

Algoritmo k-medoids

entrada: $O = \{o_1, o_2, \ldots, o_n\}$
k: número de *cluster*s

saída: $C = (c_1, c_2, \ldots, c_k)$

selecionar aleatoriamente *k* objetos de *O* para serem os medoides iniciais;
repetir
associar cada objeto não medoid *o* ao medoide *m* mais semelhante com base em uma função de similaridade;
para cada medoid *m* **faça**
para cada objeto não-medoid *o* **faça**
- trocar *m* e *o;*
- calcular custo total da configuração *S;*
fimpara
fimpara
selecionar a configuração com o menor custo;
até os medoides não mudarem

O custo entre dois objetos pode ser calculado conforme Equação 9, em que *x* é qualquer objeto, *c* é o medoide e *d* é a dimensão do objeto. O custo total da configuração *S* é a soma dos custos de todos os objetos de um *cluster* ao seu medoide.

$$Cost(x, c) = \sum_{i=1}^{d} |x_i - c_i| \tag{9}$$

3.3.3 Avaliação do Conhecimento

A avaliação do conhecimento pode ser feita por índices estatísticos que indicam a qualidade dos resultados (Rezende et al., 2011). O uso de algum índice estatístico para avaliação do resultado de agrupamento é importante, pois os algoritmos podem apresentar grupos forte e fracamente coesos. Portanto, critérios de avaliação são importantes para

fornecer aos usuários um grau de confiança para os resultados de agrupamento derivados dos algoritmos utilizados (Xu & Wunsch, 2005).

Segundo Xu e Wunsch (2005), os critérios para avaliação de resultado de agrupamento podem ser divididos em: (i) *critérios internos,* que são baseados em informações do próprio conjunto de dados e não são dependentes de informações externas (conhecimento prévio), (ii) *critérios externos,* que são baseados em alguma estrutura predefinida, que é o reflexo de uma informação prévia sobre os dados; e (iii) *critérios relativos,* que se baseiam na comparação de diferentes resultados de agrupamento, de modo a se ter uma referência para decidir qual deles melhor revela as características dos dados.

Na literatura, existem diversos índices para validação de resultados de agrupamento. Em Milligan e Cooper (1985), são comparados 30 índices de avaliação de agrupamento para determinar o número adequado de *cluster*s em um conjunto de dados. Em Vendramin et al. (2010), os autores comparam 40 índices de critério relativo de avaliação de agrupamentos. Uma revisão das abordagens de validação de resultado de agrupamento baseadas em critérios internos e externos pode ser vista em Halkidi et al. (2002a) e uma revisão de abordagens baseadas em critérios relativos pode ser vista em Halkidi et al. (2002b).

Proposto por Rousseeuw (1987), o coeficiente de Silhueta é um índice de critério relativo que aplica a ideia de coesão e separação para indicar o quão bem um objeto foi colocado em seu *cluster*. O trabalho de Vendramin et al. (2010) comparou 40 critérios de validação de agrupamento e concluiu que o coeficiente de Silhueta foi o critério que apresentou o melhor desempenho em diferentes cenários de avaliação. Em se tratando de agrupamento de textos, o valor do coeficiente de Silhueta para um documento d_i pode ser calculado conforme descrito na Equação 10a:

$$s(\mathrm{d_i}) = \frac{b(\mathrm{d_i}) - a(\mathrm{d_i})}{\max\{\mathrm{a(d_i)}, \mathrm{b(d_i)}\}} \tag{10a}$$

em que $a(d_i)$ é a dissimilaridade média entre d_i e todos os documentos de seu grupo e $b(d_i)$ é a dissimilaridade média entre d_i e todos os outros documentos do grupo vizinho mais próximo a ele. O valor do coeficiente de Silhueta fica entre -1 e 1 conforme descrito na Equação 10b:

$$s(\mathrm{d_i}) = \begin{cases} 1 - a(\mathrm{d_i})/b(\mathrm{d_i}) & se\ a(\mathrm{d_i}) < b(\mathrm{d_i}), \\ 0 & se\ a(\mathrm{d_i}) = b(\mathrm{d_i}), \\ b(\mathrm{d_i})/a(\mathrm{d_i}) - 1, & se\ a(\mathrm{d_i}) > b(\mathrm{d_i}) \end{cases}$$

(10b)

Um valor próximo de 1 indica que o documento d_i foi bem classificado em seu grupo, ou seja, $a(d_i) < b(d_i)$, o que mostra que o documento d_i está mais próximo dos documentos de seu grupo. Logo, quanto mais próximo de 1 for o valor do coeficiente de Silhueta, melhor é a qualidade do agrupamento. Já um valor próximo de -1 indica que o documento d_i foi mal classificado, ou seja, $a(d_i) > b(d_i)$, o que mostra que o documento d_i, em média, está mais distante dos documentos de seu grupo do que dos documentos do grupo vizinho mais próximo a ele. Por fim, um valor próximo de 0 indica que o documento d_i está em um ponto intermediário entre seu grupo e grupo vizinho mais próximo a ele.

O coeficiente de Silhueta para um resultado de agrupamento pode ser obtido calculando-se a média das silhuetas de cada documento, obtendo-se o valor global de Silhueta de uma solução. O valor global de Silhueta pode ser obtido conforme descrito na Equação 11.

$$Silhueta(\mathrm{S}) = \frac{1}{n}\sum_{i=1}^{n} s(\mathrm{d_i})$$

(11)

em que n é o número de documentos em uma coleção.

Segundo Kaufman e Rousseeuw (1990), a interpretação do resultado do agrupamento por meio do valor global de Silhueta pode ser feito a partir de quatro intervalos: (i) entre 0,71 e 1,0 indica que uma forte estrutura de agrupamento foi encontrada; (ii) entre 0,51 e 0,70 indica que uma razoável estrutura de agrupamento foi encontrada; (iii) entre 0,26 e 0,50 indica que uma fraca estrutura de agrupamento foi encontrada e pode ser superficial necessitando de método adicional para análise dos dados; e, (iv) menor que 0,25 indica que nenhuma estrutura de agrupamento foi encontrada.

O coeficiente de Silhueta permite comparar partições obtidas por diferentes algoritmos de agrupamento e diferentes números de grupos (Rezende et al., 2011). Ele pode

ser utilizado para estimar o melhor número de grupos para o resultado de agrupamento, sendo uma alternativa válida para determinar o valor de *k* para algoritmos como *k-means* e *k-medoids.*

A seção a seguir apresentará alguns problemas da engenharia de requisitos que estão sendo resolvidos por técnicas de mineração de textos.

3.4 Problemas da Engenharia de Requisitos Solucionados por Técnicas de Mineração de Textos

Pesquisas recentes têm utilizado técnicas de mineração de textos para contribuir para a análise e compreensão de requisitos de *software* escritos em linguagem natural. O uso da mineração de textos na engenharia de requisitos vem contribuindo em tarefas, tais como: separação e identificação de requisitos funcionais, rastreamento de requisitos, identificação de interesses transversais, entre outras. (Casamayor, Godoy, & Campo, 2011).

O sucesso ou fracasso de um projeto de *software* está relacionado à fase de engenharia de requisitos (Sateli et al., 2012). A falta de conformidade dos requisitos acarreta problemas nas fases de projeto, codificação e testes. Os documentos de especificação de requisitos podem conter trechos ambíguos uma vez que são comumente escritos em forma de textos em linguagem natural irrestrita (Hussain et al., 2009). Técnicas de classificação de textos têm sido abordagens utilizadas para identificar ambiguidades textuais nos documentos de especificação de requisitos (Hussain et al., 2007; Hussain et al., 2009).

A priorização de requisitos consiste na seleção de um conjunto de requisitos que atendam às necessidades e às expectativas do cliente. Infelizmente, as técnicas de priorização de requisitos existentes não apresentam resultados satisfatórios quando empregadas em sistemas de larga escala com inúmeros requisitos e *stakeholders* (Duan et al., 2009). Técnicas de mineração de texto têm sido utilizadas para identificar quais são os requisitos mais importantes para o cliente e que influenciam a arquitetura do sistema. Para Laurent et al. (2007), técnicas de agrupamento de texto podem organizar os requisitos em categorias, facilitando a análise e a geração de lista priorizada de requisitos. O agrupamento de texto em combinação com outras técnicas revelou-se viável para auxiliar processos de elicitação e

priorização de requisitos em sistemas que envolvem milhares de *stakeholders* (Cleland-Huang & Mobasher, 2008).

Os requisitos de *software* podem estar inter-relacionados ou mesmo relacionados a artefatos produzidos nas fases seguintes à engenharia de requisitos. O rastreamento de requisitos é uma importante tarefa da engenharia de requisitos, pois permite acompanhar ligações (*links*) estabelecidas entre requisitos, e entre os requisitos e outros artefatos. O esforço necessário para estabelecer e manter os *links* entre os requisitos tem incentivado o surgimento de pesquisas que propõem ferramentas semi-automáticas para auxiliar essa tarefa. Técnicas de mineração de texto têm sido utilizadas como partes de abordagens para verificar a exatidão e a integralidade da matriz de rastreabilidade de requisitos (Port et al., 2011), para gerar *links* candidatos entre os requisitos (Niu & Mahmoud, 2012), para auxiliar a interpretação e avaliação dos *links* de artefatos de *software* (Duan & Cleland-Huang, 2007b), e para obter conhecimento por meio de *links* já estabelecidos (Gervasi & Zowghi, 2011), entre outras.

Os requisitos de *software* também podem se relacionar de forma transversa formando os chamados interesses transversais. Nesse tipo de relacionamento, um requisito A "cruza" o requisito B de forma que esse último não pode ser desenvolvido sem que o requisito A seja considerado (Rosenhainer, 2004). Os interesses transversais podem ser identificados na fase de elicitação e análise dos requisitos. A identificação de interesses transversais nas fases iniciais do processo de desenvolvimento de *software* proporciona uma melhor modularização do sistema, além de prover informações úteis para o projeto da arquitetura do sistema. Existem algumas abordagens para a identificação semi-automática de interesses transversais em nível de requisitos. Elas podem utilizar técnicas de recuperação de informação (Rosenhainer, 2004), processamento de linguagem natural (Sampaio et al., 2005) e técnicas automatizadas de agrupamento de textos (Duan & Cleland-Huang, 2007a).

A modularização de *software* é uma importante e desejável propriedade para um sistema computacional. A modularização permite dividir o sistema em partes menores, de forma que o sistema fique mais organizado, com partes mais reaproveitáveis e com maior manutenabilidade. Apesar do estudo de modularização fazer parte da fase de projeto da arquitetura de *software*, os requisitos de *software* podem prover informações úteis que sugerem possíveis pontos de decomposição. Técnicas de mineração de texto podem ser

utilizadas para sugerir um projeto inicial do sistema com base em agrupamento de requisitos funcionalmente semelhantes (Al-Otaiby, AlSherif, & Bond, 2005).

Após a coleta dos requisitos iniciais, é necessário estimar o tamanho do sistema. Conhecer o tamanho do sistema a partir dos requisitos é uma tarefa complexa mas fundamental para determinar o esforço, tempo de desenvolvimento, cronograma e custos do projeto. Técnicas de mineração de texto podem ser utilizadas como parte de um processo de estimação de tamanho do sistema. A classificação de requisitos, por exemplo, permite organizar os requisitos em classes previamente definidas de processos funcionais com base no tamanho dos requisitos (Hussain et al., 2010).

Neste capítulo foram apresentados conceitos referentes a mineração de textos. O agrupamento de texto é uma tarefa da mineração de textos que permite extrair conhecimento a partir do agrupamento de textos similares. Por esta razão, a tarefa de agrupamento de textos é uma abordagem viável para análise de requisitos textuais e pode ser utilizada para solucionar problemas da engenharia de requisitos. O algoritmo de agrupamento *k-medoids*, as medidas de similaridade *Cosine Similarity* e coeficiente de Jaccard, o modelo de espaço vetorial e o coeficiente de Silhueta, assuntos vistos neste capítulo, foram considerados para a definição da metodologia utilizada neste trabalho.

4 Similaridade Semântica

Existem algumas medidas ou funções que podem ser utilizadas para verificar a similaridade entre textos. Uma das abordagens para verificar a semelhança entre textos é basear-se em dados estatísticos, como por exemplo, o número de vezes em que uma palavra aparece. Outra estratégia é basear-se no papel que uma palavra possui em uma sentença ou no seu significado. Como cada medida de similaridade possui uma estratégia diferente para análise de textos, ela pode apresentar diferentes resultados dependendo do texto que está sendo analisado.

As medidas de similaridade semântica são utilizadas para calcular a similaridade entre conceitos (ou termos) que não são lexicograficamente semelhantes (Varelas et al., 2005) e que estão incluídos em fontes de conhecimento (Slimani, 2013). As medidas de similaridade semântica têm sido utilizadas para a resolução de problemas de áreas como processamento de linguagem natural, recuperação de informação, segmentação de textos, sistemas recomendadores, extração de informações, etc. A abordagem típica para encontrar a semelhança semântica entre dois termos é o uso de um método que explore uma base de dados lexical (p. ex. WordNet), para produzir uma pontuação de similaridade com base nas relações existentes entre as palavras. Neste capítulo, será introduzida a base de dados lexical WordNet (Fellbaum, 1998) e serão apresentadas algumas medidas de similaridade semântica e as abordagens utilizadas por cada uma delas.

4.1 Bases de Dados Lexicais

Uma base de dados lexical é um banco de dados que contém todos os léxicos de um idioma, ou seja, é um conjunto de palavras utilizadas por um idioma. As bases de dados lexicais armazenam as definições das palavras, seus sinônimos, sua classe lexical, informações referentes ao contexto da palavra, assim como as relações semânticas entre elas. Para o idioma Português, podemos citar como exemplos, a base de dados lexical DIADORIM (Greghi et al., 2002) e WordNet.Br (Dias e Moraes, 2002). Para o idioma Inglês podemos citar a base lexical DANTE (Kilgarriff, 2010) e a base lexical WordNet (Miller, 1995).

A WordNet é uma grande base de dados lexical para o idioma Inglês, produto de um projeto de pesquisa da Universidade de Princeton. A WordNet é uma das bases de dados lexicais mais utilizadas em pesquisas envolvendo análise de textos e foi inspirada em teorias psicolinguísticas e computacionais de memória lexical humana (Fellbaum, 1998). Nesse importante banco de dados, os nomes, verbos, advérbios e adjetivos são agrupados dentro de conjuntos de sinônimos cognitivos chamados *synsets*.

Os *synsets* são interligados por relações conceituais semânticas e lexicais. Na WordNet, a principal relação entre as palavras é a relação por sinônimos, em que as palavras denotam o mesmo conceito e que são permutáveis em muitos contextos. Existem quatro relações semânticas para nomes utilizadas na WordNet, que são hipônimo/hiperônimo (*is-a*), merônimo/homônimo (*part-of*), parte merônimo/parte holônimo (*member-of*) e substância merônimo/substância holônimo (*substance-of*) (Meng et al., 2013). Na WordNet, a relação do tipo hipônimo/hiperônimo (*is-a*) é a mais comuns, correspondendo a cerca de 80% do total das relações. A Figura 9 apresenta um fragmento de relação *is-a* entre conceitos na WordNet.

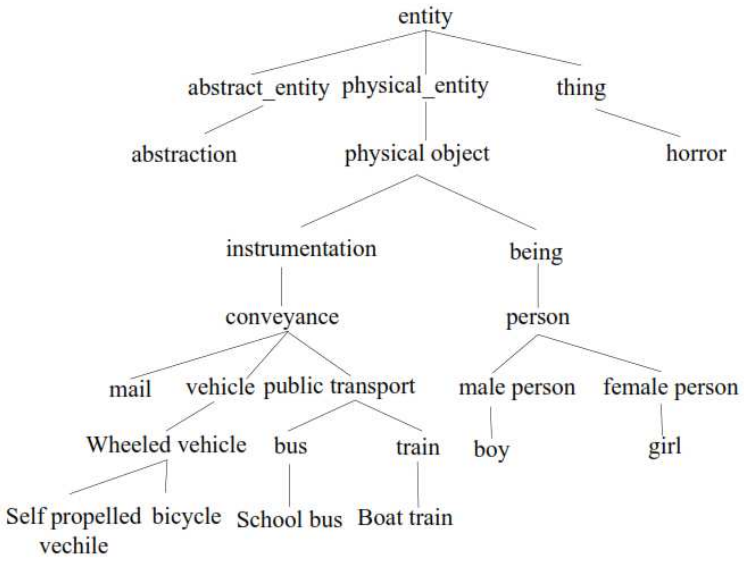


Figura 9 - Fragmento de relação do tipo *is-a* (Meng et al., 2013)

A WordNet, como uma base de conhecimento, pode ser representada como um grafo, em que cada nó representa um *synset* e cada aresta direcionada $v \rightarrow w$ indica que w é um hiperônimo de v. A Figura 10 apresenta, na forma de grafo, o fragmento de relação *is-a* visto na Figura 9.

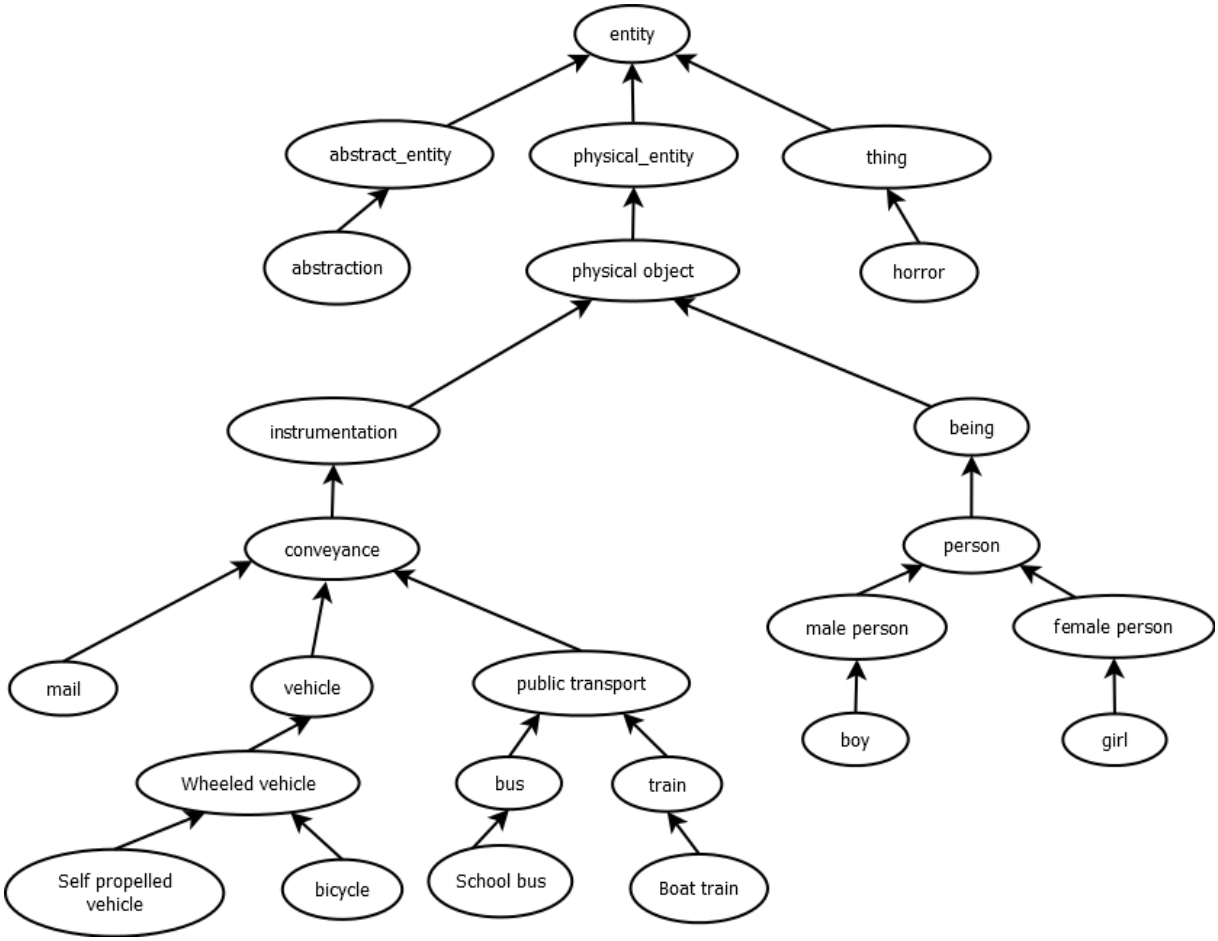


Figura 10 - Fragmento de relação do tipo *is-a* na forma de grafo
(adaptada de Meng et al., 2013)

A versão 3.1 da WordNet apresenta 155.287 palavras , 117.659 *synsets* e 206.941 pares palavra-sentido organizadas em hierarquias taxonômicas. Alguns dos métodos de semelhança semântica mais populares são implementados e avaliados usando a WordNet como ontologia de referência (Varelas et al., 2005). A seguir, será introduzido o funcionamento de funções de similaridade que fazem uso do banco de dados lexical WordNet.

4.2 Medidas de similaridade semântica baseadas na WordNet

As palavras podem apresentar relações entre si. Por exemplo, as palavras *cão* e *formiga*, apesar de diferentes estão relacionadas entre si, pois a palavra *cão* é hipônimo de *animal,* assim como a palavra *formiga*. As medidas de similaridade semântica entre conceitos são um método para medir a similaridade semântica ou distância semântica entre dois conceitos para uma dada ontologia (Slimani, 2013).

O objetivo das medidas de similaridade semântica é extrair automaticamente um valor que indica a semelhança entre as palavras mediante correspondências lexicais simples. Elas utilizam a informação encontrada em uma hierarquia *is-a* de conceitos (ou *synsets*) para quantificar o quanto o conceito *A* é semelhante a um conceito *B* (Pedersen et al., 2004).

As medidas de similaridade semântica baseadas na WordNet podem utilizar abordagens distintas, como, por exemplo, a abordagem baseada no caminho, a abordagem baseada no conteúdo da informação e a abordagem baseada em definições. Nas próximas seções, serão apresentadas tais abordagens, assim como algumas medidas de similaridade semântica pertencentes a elas.

Para uma melhor compreensão das medidas de similaridade que serão descritas nas próximas seções, seguem algumas definições de acordo com Meng et al. (2013, 2014):

(1) O comprimento do caminho mais curto de um conceito c_i até o conceito c_j é denotado por $len(c_i,c_j)$ e é calculado a partir da quantidade de arestas ou de nós, por exemplo, na Figura 9, *len*(*bus,train*) = 2;

(2) O conceito comum mais específico que os conceitos c_i e c_j compartilham (ou *most specific common subsumer*) é representado por $lso(c_i, c_j)$, por exemplo, na Figura 9, *lso*(*bus,train*) é *public transport;*

(3) A profundidade de um conceito c_i é denotado por $depth(c_i)$ e é o comprimento do caminho partindo do conceito raiz *entity* até c_i, por exemplo, na Figura 9, *depth*(*bus*) = *len*(*entity,bus*) = 7, em que *depth*(*entity*) = 1;

(4) O *deep_max* refere-se à profundidade máxima da taxonomia, por exemplo, na Figura 9, o *deep_max* é 8;

(5) O número de hipônimos para um dado conceito c_i é denotado por $hypo(c_i)$;

(6) A similaridade semântica entre os conceitos c_i e c_j é representado por $sim(c_i,c_j)$.

De acordo com Meng et al. (2013), o comprimento do caminho mais curto entre os conceitos c_i e c_j em uma taxonomia, como na Figura 9, pode ser determinada por um dos três casos a seguir:

Caso 1: $len(c_i,c_j) = 0$ quando c_i e c_j são os mesmos conceitos, sendo que c_i, c_j e $lso(c_i, c_j)$ são o mesmo nó;

Caso 2: $len(c_i,c_j) = 1$ quando c_i e c_j não são o mesmo nó, mas c_i é pai de c_j e;

Caso 3: $1 < len(c_i,c_j) <= 2*deep\_max$ quando nem c_i e c_j são os mesmos conceitos nem c_i é pai de c_j.

A seguir, serão apresentadas algumas medidas de similaridade semântica baseadas no comprimento do caminho.

4.2.1 Medidas de similaridade baseadas no caminho entre os conceitos

Em um banco de dados em que os conceitos são organizados de forma hierárquica é intuitivo imaginar que conceitos próximos são mais similares que conceitos que estão hierarquicamente afastados. Nessa abordagem, a similaridade entre dois conceitos é uma função que mede o comprimento do caminho de ligação dos conceitos e a posição dos conceitos na taxonomia. A Figura 11 ilustra esta abordagem, em que se mostra o comprimento do caminho do termo *nickel* até *coin* (1), *dime* (2), *money* (5) e *Richter scale* (7).

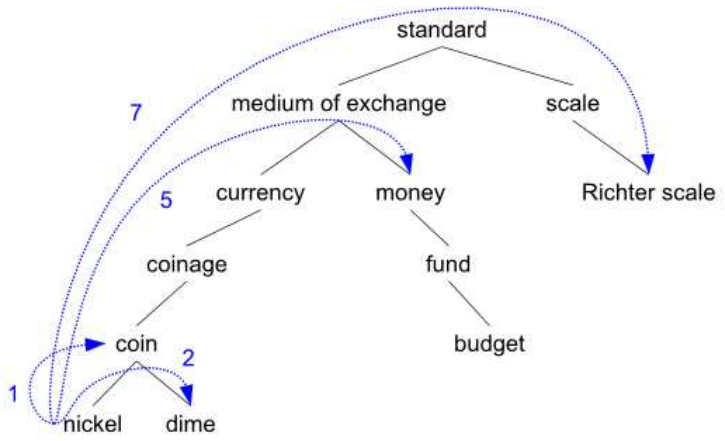


Figura 11 - Um framento da WordNet mostrando o comprimento entre caminhos (Jurafsky & Martin, 2009)

A distância semântica é um modo intuitivo de avaliar a similaridade em uma rede semântica hierárquica. A medida de similaridade *Path* é uma variação do método baseado em distância semântica proposto por Rada (Rada et. al, 1989), em que a similaridade se dá pela contagem do número de arestas existentes ao longo do caminho mais curto entre os conceitos. Segundo Rada et al. (1989), quando apenas relações *is-a* são usadas em uma rede semântica, os termos *relacionamento semântico* e *distância semântica* são equivalentes e pode-se usar o último para definir o primeiro. Nessa abordagem, a distância conceitual entre dois nós é proporcional ao número de arestas que separam os dois nós na hierarquia. A similaridade semântica entre os conceitos c_1 e c_2 em uma estrutura hierárquica é dada pela Equação 12:

$$sim_{path}(c_1, c_2) = 2 * deep\_max - len\,(c_1, c_2) \tag{12}$$

em que *deep_max* é um valor fixo para uma versão específica da WordNet.

Na Equação 12, a similaridade entre dois conceitos c_1 e c_2 é o caminho mais curto $len(c_1,c_2)$ de c_1 até c_2. Se $len(c_1,c_2)$ é 0, $sim_{path}(c_1,c_2)$ obtém o valor máximo de 2**deep_max*. Se $len(c_1,c_2)$ é 2**deep_max*, $sim_{path}(c_1,c_2)$ obtém o valor mínimo 0. Dessa forma, os valores de $sim_{path}(c_1,c_2)$ estão entre 0 e 2**deep_max*, sendo que, valores próximos a 2**deep_max* representam maior similaridade.

A medida de similaridade WuP (Wu & Palmer, 1994) é uma medida com base na profundidade de conceitos presentes em uma estrutura de rede semântica que relaciona os conceitos, ou seja, uma ontologia lexical, como, por exemplo, a WordNet. Sejam c_1 e c_2 dois conceitos em uma taxonomia, a similaridade WuP considera a posição de c_1 e c_2 até a posição do conceito comum mais específico *C (most specific common subsumer)* (Slimani, 2013). O *Lowest Super-Ordinate* (LSO ou *most specific common subsumer*) de dois conceitos c_1 e c_2 é o mais próximo conceito compartilhado por c_1 e c_2 em que a relação entre eles seja do tipo *is-a*.

Tomemos como exemplo a hierarquia de conceitos apresentada na Figura 12. Podemos dizer que um *carro* é um *automóvel* e um *automóvel* é um *veículo*. Também podemos dizer que um *avião* é um *veículo* e que um *veículo* é um *objeto*. Nesse caso, *automóvel* é pai (e também antepassado) de *carro* e que *veículo* é o antepassado de *carro* e também de *avião*. Assim, o LSO de *carro* e *avião* é *veículo*, já que é o conceito mais específico que é antepassado de *carro* e *avião*. Observa-se que *objeto* é o *common subsumer* de *carro* e *avião*, mas não é o mais próximo, já que *objeto* possui um filho (*veículo*) que também é um *common subsumer* de *carro* e *avião*.

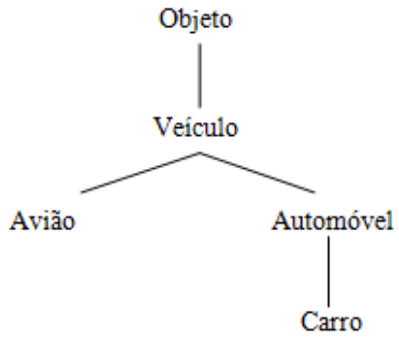


Figura 12 - Exemplo de hierarquia de conceitos do tipo é *is-a*

A similaridade de WuP pode ser calculada conforme Equação 13:

$$sim_{wup}(c_1, c_2) = \frac{2 * N}{N1 + N2 + 2 * N} \quad (13)$$

em que *N1* e *N2* são o número de nós (ligações do tipo *is-a*) que separa, respectivamente, os conceitos c_1 e c_2 do conceito comum mais próximo (LSO) entre c_1 e c_2, e *N* é o númeero de nós que separa o conceito comum mais próximo entre c_1 e c_2 do nó raiz. O valor produzido pela medida de similaridade WuP varia entre 0 e 1.

A medida de similaridade LCH (Leacock & Chodorow, 1998) utiliza relações de generalização (hiperônimo) e especialização (hipônimo) para encontrar o menor caminho entre pares de conceitos. A medida de similaridade LCH utiliza a profundidade da taxonomia em que os *synsets* são encontrados, podendo ser influenciada pela presença ou ausência de um único nó raiz. Se houver um único nó, existirão apenas duas taxonomias: uma para substantivos e uma para verbos. Dessa forma, todos os nomes estarão em uma taxonomia e todos os verbos em outra taxonomia. A Equação 14 descreve o cálculo da medida de similaridade LCH.

$$sim_{LCH}(c_1, c_2) = -log \frac{len(c_1, c_2)}{2 * deep\_max} \tag{14}$$

O valor *deep_max* é o maior caminho de um *synset* mais geral a um mais específico, ou seja, é a profundidade máxima que pode ter a árvore taxonômica da WordNet. Dependendo da versão da WordNet, o valor para *deep_max* será algo em torno de 18 e 14 para a taxonomias substantivo e verbos respectivamente. Quando c_1 e c_2 têm o mesmo sentido, *len*(c_1,c_2) é zero. Para evitarmos *log*(*0*), costuma-se adicionar 1 para *len*(c_1 e c_2) e para 2 * *deep_max*. Dessa forma, o valor da similaridade LCH estará entre 0 e *log*(*2*deep_max+1*), inclusive.

A medida de similaridade Li (Li et al., 2003) é uma medida derivada intuitivamente e empiricamente que combina o caminho mais curto entre dois conceitos e a profundidade em uma taxonomia do conceito mais específico, comum em uma função não linear. O cálculo da medida de similaridade Li é descrito conforme Equação 15:

$$sim_{Li}(c_1, c_2) = e^{-\alpha * len(c_1, c_2)} \frac{e^{\beta * depth(lso(c_1, c_2))} - e^{-\beta * depth(lso(c_1, c_2))}}{e^{\beta * depth(lso(c_1, c_2))} + e^{-\beta * depth(lso(c_1, c_2))}} \tag{15}$$

em que $\alpha \geq 0$ e $\beta \geq 0$ são parâmetros que dimensionam a contribuição do caminho mais curto e da profundidade, respectivamente. Segundo Li et al. (2003), os melhores valores são $\alpha = 0.2$ e $\beta = 0.6$. O valor da medida de similaridade Li está entre 0 e 1, sendo 1 o valor dado a conceitos totalmente similares.

A seguir, serão apresentadas algumas medidas de similaridade semântica baseadas no conteúdo da informação.

4.2.2 Medidas de similaridade baseadas no conteúdo da informação

O Conteúdo da Informação ou IC (do Inglês, *Information Content*) é uma medida de especificidade para um conceito e é calculada com base na frequência de ocorrência de um conceito em um texto (Pedersen, 2010). As medidas de similaridade baseadas no conteúdo da informação procuram, dentro de uma taxonomia, o conceito generalizador comum mais específico que liga dois conceitos. Dessa forma, valores altos de IC estão associados a conceitos mais específicos (p. ex. "caneta"), enquanto que baixos valores de IC referem-se a conceitos mais gerais (p. ex. "ideia").

Em termos matemáticos, para um conceito *c* em uma taxonomia, *P(c)* é a probabilidade de encontrar uma instância do conceito *c*. Um conceito com alta probabilidade de ocorrência significa que se trata de um conceito mais geral, pois tem sua probabilidade acrescida de todos os seus conceitos específicos. Assim, quanto maior a probabilidade de ocorrência de um conceito, mais abstrato ele é, e, consequentemente, menor é o conteúdo da informação que o conceito possui.

De acordo com a definição padrão encontrada na teoria da informação, o conteúdo da informação de um conceito *c* é representado pela Equação 16.

$$IC(c) = -\log P(c) \tag{16}$$

A probabilidade de um conceito *c* pode ser estimada conforme Equação 17.

$$P(c) = \frac{freq(c)}{S} \tag{17}$$

em que *S* é o número total de substantivos e *freq*(*c*) é a frequência da instância do conceito *c* na taxonomia. Em *freq*(*c*), cada substantivo que ocorreu no *corpus* é contado como uma ocorrência de cada classe taxonômica que a contém. Por exemplo, na Figura 13, uma ocorrência do substantivo *dime* pode ser contado com base na frequência de *dime*, *coin* e assim por diante. Desta forma, a frequência da instância de um conceito *c* pode ser calculada conforme Equação 18.

$$freq(c) = \sum_{n \in W(c)} count(n) \tag{18}$$

em que *W*(*c*) é o conjunto de palavras subordinadas ao conceito *c*.

A Figura 13 apresenta um fragmento da taxonomia WordNet. As linhas sólidas representam relações do tipo *is-a*; enquanto que, as linhas tracejadas indicam nós intermediários que foram omitidos para economizar espaço.

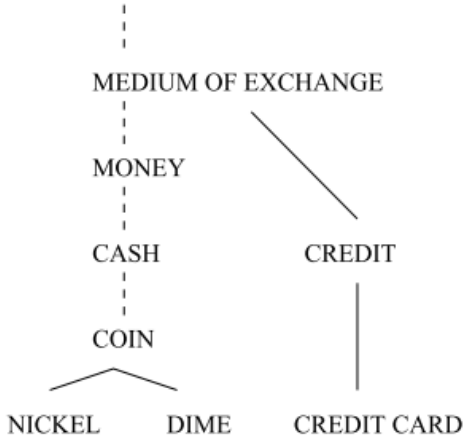


Figura 13 - Fragmento da taxonomia WordNet (Resnik, 1995)

Na Figura 13 vemos o conceito *coin* como o conceito generalizador mais específico de *nickel* e *dime*, de forma que, os conceitos *nickel* e *dime* estão subordinados ao conceito *coin*. Já, *nickel* e *credit_card* são generalizados pelo conceito comum mais específico *medium_of_exchange.*

A medida de similaridade proposta por Resnik (Resnik, 1995) utiliza o conteúdo da informação dos conceitos. Na medida de similaridade Resnik, a similaridade entre dois conceitos é dada pela quantidade de informação comum que os termos compartilham; ela é indicada pelo conteúdo de informação do conceito generalizador mais específico (LSO), que inclui ambos os conceitos em uma taxonomia.

Segundo Resnik, a similaridade semântica entre um par de conceitos c_1 e c_2 pode ser definida conforme Equação 19. O valor da medida de similaridade de Resnik será sempre maior ou igual a zero, em que zero representa conceitos sem similaridade. Na verdade, o limite superior dependerá do tamanho do *corpus* utilizado para calcular o conteúdo da informação e pode ser calculado como *ln*(*NP*) em que *NP* é o número de palavras no *corpus*. Para calcular a probabilidade de ocorrência dos substantivos na WordNet, Resnik utilizou o *Brown Corpus of America English* (Francis, Kučera, & Mackie, 1982).

$$sim_{Resnik}(c_1, c_2) = -log\ p\big(lso(c_1, c_2)\big) = IC(lso(c_1, c_2)) \tag{19}$$

A Figura 14 apresenta uma hierarquia de substantivos do WordNet. O número atribuído a cada nó representa o conteúdo da informação, conforme Equação 19. De acordo com o método proposto por Resnik, a similaridade entre os conceitos *car* e *bicycle* é o conteúdo da informação do conceito generalizador mais específico *vehicle*, no caso 8.30. A similaridade entre os conceitos *car* e *fork* é o conteúdo da informação do conceito *Artifact,* no caso 3.53. Esses resultados mostram que *car* e *bicycle* são mais similares que *car* e *fork,* o que realmente é verdade.

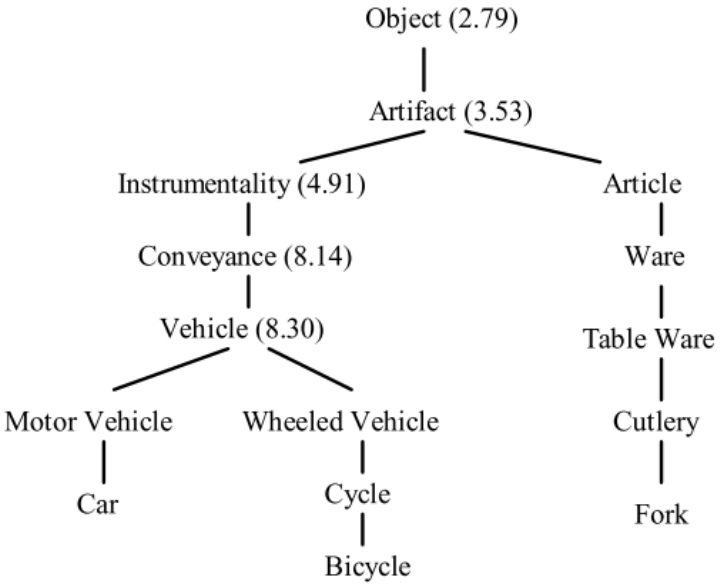


Figura 14 - Fragmento da hierarquia de substantivos da WordNet (Jiang & Conrath, 1997)

Entretanto, o método proposto por Resnik não distingui a similaridade semântica de pares de conceitos que possuam o mesmo conceito generalizador. Por exemplo, a similaridade semântica entre os conceitos *bicycle* e *fork* é a mesma que *bicycle* e *article*.

Para resolver tal problema, Jiang e Conrath (Jiang & Conrath, 1997) propõem uma medida de similaridade semântica a qual combina a abordagem baseada na distância semântica (seção anterior) e a abordagem baseada no cálculo do conteúdo da informação. O cálculo da similaridade semântica proposto pelos autores é descrito na Equação 20.

$$dis_{Jiang}(c_1, c_2) = \big(IC(c_1) + IC(c_1)\big) - 2 * IC(lso(c_1, c_2))$$

(20)

Os autores exemplificam o uso dessa medida de dissimilaridade dentro de um espaço multidimensional semântico (no caso, uma hierarquia de substantivos como a WordNet) em que cada nó (conceito) possui uma massa (conteúdo da informação) e se encontra em um eixo específico. Dessa forma, a distância semântica entre dois conceitos que estão no mesmo eixo é apenas a diferença entre as massas semânticas, enquanto que a distância semântica de conceitos em eixos é a soma das duas distâncias em relação a um

conceito generalizador comum. O valor produzido pela medida de similaridade proposta por Jiang e Conrath varia entre 0 e 1.

O trabalho de Lin (Lin, 1998) apresenta uma definição formal sobre similaridade e propõe uma medida de similaridade genérica. Lin define uma medida de similaridade que é universal (não atrelada a qualquer forma de representação do conhecimento) e teoricamente justificada ("derivada de um conjunto de pressupostos" ao invés "derivada de uma fórmula"). A medida proposta por Lin utiliza as seguintes intuições como base:

1. A similaridade entre os conceitos A e B está relacionada à **comunalidade** entre eles. Quanto maior for a **comunalidade** que eles compartilham, mais similares eles são;
2. A similaridade entre os conceitos A e B está relacionada à **diferença** entre eles. Quanto mais diferenças os conceitos possuírem, menos similares eles são;
3. A similaridade máxima entre os conceitos A e B é alcançada quando A e B são idênticos, não importando quanta **comunalidade** eles compartilham.

Lin define **comunalidade** entre os conceitos *A* e *B* como a quantidade de informação da proposição que indica os pontos em comum entre eles e é calculada conforme Equação 21.

$$IC(comm(A,B)) \tag{21}$$

e a diferença entre A e B é calculada conforme Equação 22.

$$IC\big(descr(A,B)\big) - IC(comm(A,B)) \tag{22}$$

em que *descr(A,B)* é a proposição que descreve o que *A* e *B* são.

O valor produzido pela medida de similaridade Lin varia entre 0 e 1. Para calcular a similaridade entre dois conceitos em um domínio de taxonomia, Lin propõe a Equação 23.

$$sim_{Lin}(c_1, c_2) = \frac{2 * IC(lso((c_1, c_2)))}{IC(c_1) + IC(c_2)} \quad (23)$$

A seguir, serão apresentadas algumas medidas de similaridade semântica baseadas em definições.

4.2.3 Medidas de similaridade baseada em definições

Esse tipo de medida de similaridade utiliza as informações contidas nas definições (*glosses*) das palavras, para identificar o quão similares elas são. A abordagem baseada em definições verifica o quanto de sentido compartilhado as palavras possuem em comum. O significado das palavras é extraído a partir de dicionários conhecidos, como MRD (do Inglês, *Machine Readable Dictionaries*), que possuem informações passíveis de ser extraídas por computador, como, por exemplo, *Longman Dictionary of Contemporary English* (LDOC), *Collins English Dictionary* (CED) e WordNet.

A desambiguação do sentido da palavra (do Inglês, *Word Sense Disambiguation*) é o processo de atribuição de um significado a uma palavra em particular com base no contexto em que ocorre. Para solucionar o problema de desambiguação, Lesk (1986) propôs um algoritmo baseado em definições para medir a similaridade entre dois conceitos por meio das sobreposições (*overlap*) encontradas nas definições correspondentes a cada conceito. A ideia principal do algoritmo é selecionar o sentido no qual exista o maior número de coincidências nas palavras que se quer desambiguar.

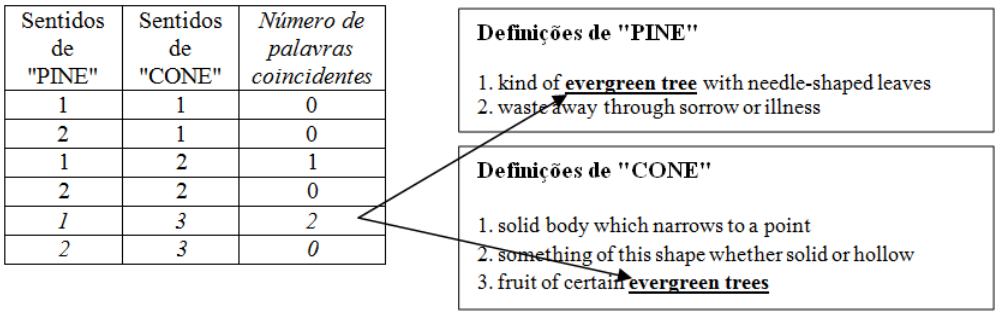
O algoritmo baseia-se na percepção de que os sentidos de palavras que estão relacionadas, muitas vezes, compartilham das mesmas palavras em suas definições. O Quadro 3 mostra o algoritmo proposto por Lesk.

Quadro 3 - Algoritmo original de Lesk

1. Extrair de um dicionário MRD todas as definições das palavras que se deseja desambiguar; 2. Contabilizar as coincidências (*overlap*) das palavras nas definições para todas as possíveis combinações dos sentidos; 3. Escolher os sentidos que possuem o maior número de coincidências.

A Figura 15 mostra o exemplo descrito no trabalho de Lesk para desambiguar a palavra *cone* na frase *pine cone,* como sugere o sub-título de seu trabalho "*how to tell a pine cone from an ice cream cone*". Neste exemplo, percebe-se que a palavra *cone* em "*ice cream cone*" refere-se a uma parte de um alimento doce, no caso, o sorvete; enquanto que, em "*pine cone*" ela refere-se à uma parte de uma planta. Na Figura 15, as definições de *pine* e *cone* são verificadas e o número de palavras coincidentes são contabilizadas. Nota-se que *Pine* #1 ⋂ *Cone* #3 = 2, no caso, *evergreen* e *tree,* o que pode sugerir que se as palavras *pine* e *cone* aparecem juntas, o sentido mais provável é que se trata do fruto de uma árvore.

O algoritmo original proposto por Lesk extrai as definições do dicionário de *Oxford Advanced Learner's Dictionary.*



Sentidos de "PINE"	Sentidos de "CONE"	*Número de palavras coincidentes*
1	1	0
2	1	0
1	2	1
2	2	0
1	*3*	*2*
2	*3*	*0*

Figura 15 - Exemplo de desambiguação para a palavra *cone*
(traduzido e adaptado de Lesk, 1986)

Uma limitação do algoritmo original de Lesk é que, as definições de sentido descritas em um dicionário geralmente são curtas, o que leva, às vezes, a uma quantidade de vocabulário insuficiente para identificar a relação entre os sentidos. O algoritmo original de Lesk considera apenas as sobreposições entre as definições da palavra alvo e as definições das

palavras vizinhas a ela, o que, de certa forma, é um número limitado de comparações de sentidos.

O algoritmo adaptado de Lesk (S. Banerjee & Pedersen, 2002) estende essa comparação para incluir as definições das palavras que estão relacionadas com as palavras do texto que está sendo desambiguado. O algoritmo adaptado de Lesk utiliza a base de dados lexical WordNet, ao invés de um dicionário tradicional, para fazer uso das definições dos *synsets* e das relações que existem entre eles. Assim, as definições das palavras vizinhas à palavra-alvo (palavra que se quer desambiguar) são expandidas de forma a incluir também as definições das palavras que se encontram relacionadas às palavras vizinhas. Banerjee e Pedersen (2002), definem o contexto da palavra alvo como um conjunto de *n* palavras à esquerda e *n* palavras à direita da palavra alvo até o total de 2*n* palavras circundantes.

Por exemplo, suponhamos que a palavra *bank* é a palavra que se deseja desambiguar e que as palavras *money* e *tree* sejam as palavras vizinhas. O algoritmo original de Lesk verifica as coincidências entre as definições de *money* com as definições de *bank* e, em seguida, as coincidências entre as definições de *tree* e *bank*. O algoritmo adaptado de Lesk verificará as mesmas coincidências e mais as coincidências das definições dos conceitos relacionados semanticamente a *bank*, *money* e *tree* de acordo com a WordNet.

O Quadro 4 mostra o algoritmo adaptado de Lesk por Banerjee e Pedersen (2002).

Quadro 4 - Algoritmo adaptado de Lesk

1. Preparar um conjunto de palavras dentro do contexto circundante;
2. Para cada sentido da palavra, obter um conjunto de palavras de diversas relações lexicais e semânticas na WordNet, incluindo sinônimos, sentidos, frases de exemplo, hiperônimos, hipônimos, etc.;
3. Os dois conjuntos acima referidos são comparados e o sentido que dá a máxima sobreposição é escolhido.

Os valores produzidos pelos algoritmos original e adaptado de Lesk são normalizados para que fiquem entre 0 e 1. Esta normalização é feita dividindo-se o valor de similaridade obtido pelo valor máximo possível de similaridade produzido pelo algoritmo.

Este capítulo apresentou as medidas de similaridade semântica mais conhecidas na literatura. Essas medidas costumam fazer uso de uma base lexical para extrair informações que permitam indicar o quão relacionados estão dois conceitos. A análise do caminho existente entre os conceitos, o conteúdo da informação ou mesmo as definições das palavras são, em geral, as estratégias mais adotadas para se verificar o nível de similaridade semântica entre palavras.

5 Trabalhos Relacionados

Foram encontrados, na literatura, trabalhos que em parte estão relacionados ao tema desta dissertação. As propostas dos trabalhos, que serão descritos neste capítulo, possuem alguma associação com os objetivos descritos nesta dissertação e, de certa forma, contribuíram para o desenvolvimento da abordagem que foi empregada para se atingir os objetivos descritos aqui.

Neste capítulo, serão apresentados alguns trabalhos relacionados a agrupamento de requisitos, à mineração de estórias de usuário e trabalhos que fazem uso de alguma medida de similaridade para análise de requisitos. O objetivo deste capítulo não é apresentar uma relação extensa de trabalhos obtida por uma pesquisa sistemática, mas mostrar os trabalhos mais importantes que apoiaram a metodologia descrita nesta dissertação.

A busca de tais trabalhos na literatura foi feita mediante consultas às bases ACM, IEEE, Springer e Google Schoolar.

5.1 Trabalhos relacionados a agrupamento de requisitos

A complexidade das atividades da engenharia de requisitos está relacionada, muitas vezes, ao tamanho do sistema a ser desenvolvido. Para facilitar essas atividades, os requisitos coletados são agrupados e organizados em grupos coerentes. Dessa forma, algumas das dificuldades encontradas na engenharia de requisitos podem ser minimizadas por meio da aplicação de técnicas de agrupamento de requisitos (Duan, 2008).

A forma mais comum de agrupar os requisitos é por meio do uso de um modelo de arquitetura do sistema para identificar subsistemas e associar os requisitos a cada um desses subsistemas (Sommerville, 2011). Os primeiros relatos sobre agrupamento de requisitos de *software* podem ser vistos em Hsia & Gupta (1992); Hsia et al. (1996); e Hsia & Yaung (1988). Apesar destes trabalhos não empregarem uma técnica específica de mineração de textos, eles foram os primeiros a explorarem os benefícios do agrupamento sistemático de requisitos. Estes trabalhos apresentam propostas de modelo de representação além de algoritmos para realizar o agrupamento de requisitos.

Nos trabalhos de Hsia e Yaung (1988) e Hsia e Gupta (1992), o agrupamento de requisitos é utilizado para auxiliar o processo de decomposição de *software*. A principal desvantagem existente em ambos os trabalhos é que os algoritmos sugeridos estão restritos a um conjunto de instruções manuais para agrupamento de requisitos. Os algoritmos propostos nesses trabalhos não promovem um agrupamento automático de requisitos e sequer exploram técnicas de mineração de textos. Outra limitação destes trabalhos é que ambos baseiam-se em requisitos organizados em uma matriz requisito-cenário, em que cada requisito está relacionado a um único cenário, o que nem sempre é verdade quando se trata de sistemas de larga escala. Em ambos os trabalhos, a ideia básica é agrupar os requisitos de acordo com suas funcionalidades, de forma que, cada *cluster* é uma unidade operacional semi-independente. Em Hsia e Yaung (1988), um dos maiores benefícios do agrupamento de requisitos é facilitar a identificação de incrementos de sistemas e estabelecer uma tecnologia de entrega incremental na indústria de *software*. Hsia e Gupta (1992) procuram minimizar a subjetividade existente no agrupamento de requisitos em um trabalho anterior (Hsia & Yaung, 1988), em que desenvolvem um algoritmo de agrupamento que emprega o conceito de tipo abstrato de dados para entrega incremental de sistema dominante de dados ao invés da abordagem por prototipação baseada em cenários.

Li et al. (2007) encapsularam requisitos de *software* em *clusters* com base em sete dimensões (atributos) de requisitos classificados em dimensões de similaridade e dimensões de associatividade. Segundo os autores, o grau de similaridade descreve a relação entre os atributos *assunto, objeto* e *ação* de um requisito. Já o grau de associatividade entre os requisitos é baseado nos atributos *funcionalidade, qualidade, tempo* e *localização*. Após a definição dos *clusters* de requisitos, eles passam por um processo de encapsulamento que consiste na definição de interfaces externas para cada um dos *clusters*. Os autores definem essas interfaces externas como o relacionamento de um determinado *cluster* com outros *cluster*s ou atores. No trabalho de Li et al. (2007), o processo de agrupamento de requisitos um processo semi-automatizado ainda dependente de esforço manual. Outra limitação é que os requisitos a serem analisados precisam fornecer informações para todos os atributos descritos anteriormente, informações que nem sempre são possíveis de se definir, como por exemplo, a *qualidade* e o *tempo*.

Palmer e Liang (1992) propuseram uma abordagem de indexação e agrupamento de requisitos para auxiliar as tarefas relacionadas à identificação de erros, incompletude e ambiguidade nos requisitos de *software*. O algoritmo proposto pelos autores, denominado

TTC (do Inglês, *Two-Tired Clustering*), realiza duas etapas de agrupamento. O agrupamento por funcionalidade é a primeira etapa do algoritmo. Nessa etapa, o algoritmo agrupa os requisitos com verbos similares dentro de um grupo de acordo com um vocabulário de verbos (tesauro) previamente definido. Na segunda etapa do algoritmo, cada grupo de requisitos gerado na etapa anterior é subdividido por meio de um agrupamento por similaridade. Nessa etapa, os requisitos são convertidos em vetores de pesos de termos e depois comparados com o centroide (ponto médio) de cada *cluster*. A similaridade entre os requisitos é obtida por meio do cosseno do ângulo formado entre seus respectivos vetores. Após a inserção do requisito em um grupo, o *cluster* poderá ser dividido, caso atinja um determinado critério. O centroide do *cluster* é recalculado a cada nova inserção de um requisito. Um limitação deste trabalho está relacionada a função de similaridade adotada já que ela é sensível a quantidade de texto a ser analisada. Para requisitos com pouca informação textual (p. ex. estórias de usuário), a função de similaridade utilizada por Palmer e Liang (1992) pode não ser adequada, uma vez que a quantidade de palavras consideradas pela função pode ser insuficiente a ponto da função não conseguir indicar alguma similaridade que realmente exista.

Nota-se, em Hayes et al. (2008), o uso de técnicas de agrupamento de texto. Os autores propuseram um método denominado PREREQIR (do Inglês, *Pre-Requirements Information Retrieval*) que realiza a recuperação, estruturação e rotulagem de pré-requisitos obtidos pelos *stakeholders,* para verificar se os envolvidos compartilham um entendimento comum do sistema. Um dos objetivos do trabalho é identificar os pré-requisitos (especificação genérica de requisito) e organizá-los em grupos comuns e incomuns (*outliers*) de necessidades de usuários. Para a tarefa de agrupamento, os pré-requisitos foram transformados em vetores de termos após passarem pelos processos de remoção de *stopword* (remoção de palavras muito comuns e de pouca relevância) e estemização (transformação das palavras em sua forma base por meio da remoção de plurais, gerúndios e sufixos). Em seguida, os vetores são comparados por meio do cosseno do ângulo formado entre eles, e, depois, agrupados com base em um limiar de similaridade. Os autores utilizaram o algoritmo de agrupamento hierárquico aglomerativo denominado AGNES (do Inglês, *AGglomerative NESting*), pelo qual foi possível evidenciar uma forte estrutura hierárquica entre os pré-requisitos. Primeiramente, os autores utilizaram o algoritmo particional PAM, mas o resultado foi um conjunto de *clusters* fracamente separados. Para os autores, o agrupamento dos pré-requisitos por *stakeholders* similares permite a análise sob o ponto de vista intra e intergrupos. A comparação intragrupo indicará, por exemplo, quão similares são os pré-requisitos para todos

os usuários, permitindo verificar o nível de entendimento do sistema entre eles. A comparação intergrupo indicará quão semelhantes são os pré-requisitos de um desenvolvedor e de um usuário final, permitindo analisar o entendimento entre diferentes *stakeholders*.

No trabalho de Castro-Herrera et al. (2008), os autores propõem um *framework* para auxiliar o processo de elicitação de requisitos em projetos que envolvam a participação de muitos *stakeholders*. Esse *framework* utiliza técnica de agrupamento de texto para organizar os requisitos em grupos visando descobrir temas dominantes e transversais para a criação de tópicos para fóruns de discussão. Os grupos de requisitos gerados são, em seguida, utilizados por sistemas recomendadores para incluir os *stakeholders* em fóruns de discussão apropriados. Em Castro-Herrera et al. (2008), o pré-processamento dos requisitos é similar ao visto em Hayes et al. (2008) fazendo uso de métodos de recuperação de informação como remoção de palavras comuns (*stopwords*), redução das palavras para sua forma base (*stemming*), análise de frequência de termos e representação dos requisitos a partir de vetores de pesos de termos. Os autores utilizam o algoritmo *Bisecting k-means* (uma extensão do algoritmo *k-means*) para realizar o agrupamento dos requisitos. Utilizaram como medida de similaridade o cosseno do ângulo formado entre os requisitos. As técnicas de recuperação de informação e de agrupamento de texto para a criação de fóruns de discussão também podem ser vistas em trabalhos subsequentes (Carlos Castro-Herrera, Cleland-Huang, & Mobasher, 2009) e (Carlos Castro-Herrera, Duan, Cleland-Huang, & Mobasher, 2009).

Os documentos de requisitos de *software* são fontes importantes para identificar elementos arquiteturais do sistema. Com o objetivo de obter a decomposição funcional de um sistema, Casamayor et al. (2012), utilizaram algoritmos de agrupamento para agrupar responsabilidades identificadas nos requisitos (ações, atividades ou tarefas que o *software* deverá desempenhar). Os autores, inicialmente, analisam requisitos funcionais com métodos de processamento de linguagem natural para detectar as responsabilidades. Depois de identificadas, elas foram agrupadas utilizando alguns algoritmos de agrupamento. Quatro algoritmos de agrupamento foram utilizados e comparados na tarefa de agrupamento de responsabilidades: *Expectation-Maximization* (EM), COBWEB, X-Means e DBSCAN. Segundo Casamayor et al. (2012), todos os algoritmos produziram bons agrupamentos de responsabilidades com destaque ao algoritmo EM, que obteve resultados melhores nos quesitos pureza e entropia dos *clusters*. Para os autores, o agrupamento das responsabilidades pode revelar componentes funcionais da arquitetura do sistema.

Na engenharia de requisitos, a identificação de especificações inconsistentes ou incompletas é uma tarefa fundamental para que o *software* não contenha funcionalidades mal interpretadas ou que funcionem parcialmente. Para identificar especificações defeituosas Ferrari et al. (2012) agruparam léxica e sintaticamente os requisitos de *software*. Os autores identificaram que conjuntos de requisitos que não puderam ser agrupados adequadamente (*outliers*) tinham uma relação com requisitos anteriormente marcados como defeituosos (requisitos incompletos ou com terminologia inconsistente). Nesse trabalho, os autores utilizaram tarefas de recuperação de informação como tokenização, que se refere ao processo de divisão do texto em palavras ou termos individualizados; remoção de *stopwords,* que consiste na remoção de palavras comuns, e estemização, para a redução de cada termo à sua raiz morfológica. Para agrupar os requisitos, os autores utilizam o algoritmo WCC, descrito em Lucchese et al. (2011). O algoritmo WCC produz um grafo ponderado não direcionado $G_D = (D,E,w)$ em que D é o conjunto de requisitos que corresponde aos nós do grafo, E é o conjunto de arestas que liga os pares de requisitos e w é a função de ponderação para medir a similaridade entre cada par de requisitos. Para analisar a similaridade dos requisitos, os autores utilizaram o coeficiente de Jaccard (Jaccard, 1901). Os experimentos de Ferrari et al. (2012) revelaram que o uso combinado do agrupamento léxico e sintático aumenta a precisão do método para identificação de requisitos defeituosos, tornando-se uma abordagem complementar para a análise de requisitos.

A rastreabilidade de requisitos é uma atividade da engenharia de requisitos que consiste na identificação de relacionamentos (*links*) entre os requisitos do sistema e os requisitos e outros artefatos do sistema, tais como, códigos fonte, planos de teste, diagramas, entre outros. Um *link* de rastreabilidade liga, por exemplo, um requisito a uma classe ou a um diagrama de caso de uso. Ferramentas de rastreamento de requisitos retornam um conjunto de *links* candidatos de artefatos de *software* que serão analisados pelo usuário. Alguns trabalhos propõem a utilização de técnicas de agrupamento em ferramentas de rastreabilidade de requisitos. Duan e Cleland-Huang (2007b) utilizam um algoritmo de agrupamento para aumentar a compreensão e avaliação dos *links* entre os documentos de *software*. Os autores avaliaram e compararam três abordagens distintas de agrupamento: agrupamento hierárquico aglomerativo, agrupamento particional (*k-means*) e agrupamento hierárquico divisivo (*bisecting divisive clustering*). Cada uma das abordagens foi avaliada em função do agrupamento de três conjuntos diferentes de requisitos. Todas as abordagens de agrupamento calcularam a similaridade entre pares de artefatos com o uso de um algoritmo de rede

probabilístico. O uso de técnica de agrupamento aumentou a compreensibilidade e reduziu o esforço humano para analisar os *links* candidatos de artefatos de *software*, uma vez que puderam ser apresentados como parte de um grupo significativo. Dessa forma, um analista humano pode avaliar os *links* dos artefatos ao mesmo tempo em que avalia os artefatos semelhante*s*.

Inspirados no trabalho de Duan e Cleland-Huang (2007b), Niu e Mahmoud (2012) utilizaram técnica de agrupamento para melhorar o processo de geração de *links* entre os artefatos de *software*. Os autores partem da hipótese de que grupos corretos e incorretos de ligações de artefatos podem estar associados a grupos de alta e baixa qualidade, respectivamente. Niu e Mahmoud (2012) utilizaram os algoritmos de agrupamento *k-means*, *bisecting divisive clustering* e três variações do algoritmo de agrupamento hierárquico aglomerativo (*single-link*, *complete-link* e *average-link)* para agrupar os *links* entre os artefatos de *software*. No experimento realizado, artefatos de *software* foram organizados em um modelo de espaço vetorial. O cosseno do ângulo entre os vetores foi utilizado para verificar a similaridade entre os artefatos. Para avaliar a qualidade dos *clusters*, os autores propuseram três heurísticas (*MAX*, *AVE* e *MED*) que fornecem, respectivamente, a máxima, a média e a mediana similaridade entre os *links* do *cluster* e um *link* de consulta informado pelo usuário. Como a qualidade dos grupos depende de seus valores de revocação (porcentagem de *links* corretos que foram recuperados), os autores avaliaram as três heurísticas, monitorando os valores de revocação de cada *cluster* toda vez que *os links* de um *cluster* de baixa qualidade eram descartados. Isso permitiu selecionar a heurística que mantinha o maior índice de revocação e utilizá-la para garantir *clusters* de alta qualidade. Como os *links* de rastreabilidade em *clusters* de baixa qualidade são, em sua maioria, falsos positivos, a remoção destes *clusters* melhora a geração de *links* candidatos de artefatos de *software*. Assim, o analista humano não destinará esforços para identificar *links* falso positivos, concentrando-se apenas na validação dos *links* de rastreabilidade corretos.

Para um adequado projeto de arquitetura do sistema, é essencial a identificação e documentação de interesses transversais durante a fase de análise de requisitos. Interesses transversais são requisitos de um sistema que cruzam (afetam) diversas partes do sistema, como, por exemplo: segurança, gerenciamento de memória, entre outros. O trabalho de Duan e Cleland-Huang (2007a) utiliza técnica de agrupamento de texto para auxiliar o processo de identificação de interesses transversais. A abordagem proposta pelos autores agrupa os requisitos em dois momentos distintos. No primeiro momento, os requisitos são agrupados

por termos dominantes que podem representar características do sistema, tarefas do usuário ou possíveis interesses transversais. Em um segundo momento, os termos dominantes são removidos dos *clusters* formados inicialmente e os requisitos são novamente agrupados para identificar novos interesses transversais, nomeados de interesses recessivos ou não dominantes. Os autores utilizaram um método modificado de agrupamento hierárquico aglomerativo para reunir os requisitos. Para calcular a similaridade entre os requisitos, os autores utilizaram um modelo probabilístico baseado na distribuição e co-ocorrência dos termos dominantes. Nesse modelo probabilístico, os termos que aparecem frequentemente nos requisitos têm menos peso do que aqueles que aparecem com menos frequência. Como a maioria dos interesses transversais está dispersa entre os requisitos de *software*, o agrupamento de requisitos permite a identificação de funcionalidades semelhantes candidatas a se tornarem interesses transversais. Trabalhos que utilizam técnicas de agrupamento para identificação de interesses transversais em nível de código fonte têm sido propostos por McFadden & Mitropoulos (2012), Moldovan & Serban (2006) e Serban & Moldovan (2006).

A priorização de requisitos ajuda os envolvidos no projeto a entender quais requisitos são mais importantes e mais urgentes (Aasem, Ramzan, & Jaffar, 2010). Técnicas de agrupamento têm sido utilizadas para resolver problemas de priorização de requisitos. Laurent et al. (2007) propõem uma abordagem chamada *Pirogov,* que utiliza mineração de texto e técnicas de aprendizado de máquina para auxiliar a priorização de requisitos. *Pirogov* utiliza um algoritmo de agrupamento hierárquico para gerar um agrupamento inicial de características baseado em termos extraídos de uma coleção de requisitos. A partir do agrupamento inicial, os requisitos são priorizados individualmente, gerando uma classificação base. Interesses transversais sob a forma de requisitos não funcionais também são identificados para a construção da lista final de requisitos priorizados. O agrupamento automático utilizado pela abordagem *Pirogov* promove uma redução do número de priorização manual de requisitos. O algoritmo de agrupamento utilizado, assim como o método de cálculo da similaridade entre os requisitos, são os mesmos de Duan e Cleland-Huang (2007b). Em Duan et al. (2009), a abordagem *Pirogov* é avaliada em um novo estudo de caso. Entretanto, ambos os trabalhos dependem da análise (por vezes, subjetiva) de engenheiros de requisitos.

Os trabalhos apresentados nesta seção utilizam, de alguma forma, a técnica de agrupamento de textos para organizar os requisitos de um sistema. Com a técnica de agrupamento de textos, estes trabalhos procuram resolver problemas comuns da engenharia de

requisitos, como a modularização, a identificação de ambiguidade, a priorização de funcionalidades, a rastreabilidade de requisitos e a identificação de interesses transversais. Percebe-se que, a técnica de agrupamento de texto é comumentemente usada para trabalhar com requisitos de software descritos em linguagem natural.

A abordagem proposta neste trabalho utilizou a técnica de agrupamento de texto para auxiliar a análise de requisitos que é feita na reunião de planejamento de *sprints* do Scrum. Neste trabalho, a técnica de agrupamento de textos foi utilizada para analisar as estórias de usuário em um *sprint* de desenvolvimento Scrum e, se necessário, recomendar uma nova composição de estórias para o *sprint* em análise.

As técnicas de preprocessamento de textos tokenização, estemização e remoção de *stopwords*, apresentadas nestes trabalhos, são técnicas eficientes que permitem preparar os textos contidos nos requisitos para serem analisados por um algoritmo computacional. Estas técnicas de preprocessamento de textos foram utilizadas neste trabalho para preparar o texto das estórias de usuário para serem analisados por um algoritmo de agrupamento e por funções de similaridade.

A abordagem proposta neste trabalho utilizou o modelo de representação de textos (Modelo de Espaço Vetorial) e as funções de similaridade *Cosine Similarity* e Coeficiente de Jaccard, que foram usados em alguns trabalhos apresentados nesta seção.

5.2 Trabalhos relacionados à mineração de estórias de usuário

Estimar o esforço para o desenvolvimento de um *software* é uma tarefa crucial. Subestimar ou superestimar o esforço de desenvolvimento pode ocasionar problemas como perda de oportunidades de negócio, prejuízo para a reputação, competitividade e desempenho Abrahamsson et al., (2011).

No trabalho de Abrahamsson et al. (2011) é realizada mineração de estórias de usuário para auxiliar a estimação de esforço em projetos que utilizam a metodologia ágil de desenvolvimento. Os autores extraem informações, chamadas de preditores, de um conjunto de estórias de usuário. Esses preditores (número de caracteres existentes, presença de palavras chaves e prioridades das estórias de usuário) serão utilizados para construir e treinar um

modelo de estimação de esforço. Para os autores, será possível estimar o esforço de uma nova estória de usuário quando ela for submetida à análise pelo modelo de estimação proposto. O agrupamento de estórias de usuário também poderá auxiliar a estimação de esforço para a implementação de requisitos.

Para Pham e Pham (2011), a falta de uma visão arquitetural do sistema promove flutuações na velocidade do time de desenvolvimento. Para os autores, isso acontece porque os desenvolvedores recebem uma "pilha de estórias de usuário" sem terem uma visão prévia de como o produto final parecerá ou mesmo de como os componentes do sistema se encaixarão. Pham e Pham (2011) sugerem que as estórias de usuário sejam agrupadas, por exemplo, com base nos dados de negócios comuns. Esta forma de agrupamento das estórias de usuário permitirá separar as funcionalidades que tratam de inserções das que tratam de leitura, atualização e exclusão. Em outras palavras, através da organização de trabalho em torno de elementos de dados comuns e do conceito CRUD (*create, read, update, delete*), será possível acelerar o desenvolvimento, já que pequenas subequipes podem trabalhar, em paralelo, em estórias de usuários específicas e auto-suficientes (Pham & Van Pham, 2011). Segundo os autores, tal estratégia evitará oscilações na velocidade do time e, consequentemente, proporcionará aumento da velocidade de desenvolvimento, além de dar condições para que as atividades de desenvolvimento ocorram em paralelo. O agrupamento de estórias de usuário proposto por Pham e Pham (2011) não utiliza qualquer ferramenta ou algoritmo computacional para a automatização dessa tarefa.

A técnica chamada de *User Story Mapping* proposta por Jeff Patton (2008, 2011), sugere que as estórias de usuário sejam agrupadas para se obter um *backlog* do produto inicial além de informações para ajudar a desenvolver um plano de *releases* e *sprints*. A técnica de mapeamento de estórias de usuário nos ajuda a entender o produto que estamos construindo, a quebrar o produto em pequenos pedaços para construí-lo de forma iterativa e incremental, e efetivamente nos auxilia a planejar lançamentos de produtos mínimos viáveis (Patton, 2011).

As sugestões propostas por Pham e Pham (2011) e por Jeff Patton (2008, 2011) apresentam apenas uma forma manual de organização de estórias de usuário, não estando amparadas por qualquer ferramenta ou processo automático, e sendo altamente dependentes da experiência de um analista humano. As propostas desses autores também não utilizam indicadores utilizados em estórias de usuário, tais como: valor de negócio - que indica a importância de um requisito e ajuda a alinhar o interesse do cliente com o produto; estimativa

de esforço - unidade de tempo utilizada para indicar o tempo necessário para a implementação da estória; e prioridade - que ajuda a indicar em que ordem as estórias devem ser atendidas.

Os trabalhos apresentados nesta seção, sugerem que as estórias de usuários sejam agrupadas para extrair algum tipo de informação útil para apoiar o desenvolvimento do sistema, como por exemplo, informação para estimar, informação sobre a visão arquitetural do sistema e informação para desenvolver planos de entrega de incrementos do sistema. Em geral, os trabalhos aqui apresentados exploram o conteúdo das estórias de usuário para se levantar algum conhecimento implícito relevante.

A ferramenta proposta neste trabalho agrupa, automaticamente, as estórias de usuário, verificando a similaridade existente entre elas, e independe de esforço manual ou de experiência de um analista humano. A avaliação do analista de requisitos é feita após uma sugestão de agrupamento proposta pela ferramenta com base na verificação de similaridade existente entre as estórias de usuário. A partir do resultado do agrupamento, o analista de requisitos poderá verificar a correlação de suas estórias selecionadas para implementação apoiado pelos grupos de estórias sugeridos pela ferramenta desenvolvida neste trabalho.

5.3 Trabalhos relacionados ao uso de medida de similaridade para análise de requisitos

No trabalho de Matsuoka e Lepage (2011) é apresentada uma abordagem para a detecção de termos ambíguos em especificações de requisitos, para contribuir com a qualidade do desenvolvimento do *software*. Segundo os autores, um sistema que verifica a existência de palavras ambíguas em requisitos de *software* ajudará o analista humano a tomar uma decisão sobre o uso correto do termo. A abordagem sugerida pelos autores consiste em, a partir de um texto *T* a ser analisado, extrair os termos candidatos a palavras ambíguas *w*, restringindo-se ao termos técnicos conforme recomendações contidas na IEEE std 830 (IEEE, 1998). Em seguida são coletados da WordNet todos os significados de *w* e é feita a atribuição deles para cada sentença *S* de *T,* por meio da medida de similaridade semântica *WuP*. Após isso, os significados de *w* para cada *S* são reatribuídos por meio de agrupamento particional. Nesse processo de agrupamento particional é utilizada a medida similaridade semântica *WuP* para verificar a semelhança entre as sentenças. Por fim, as sentenças são agrupadas novamente por seus conjuntos de significados. Segundo os autores, após esse agrupamento

final, restará apenas uma única sentença *S* onde a palavra *w* é considerada ambígua. Para outras sentenças, a palavra *w* aparentemente não é ambígua.

Para resolver o problema de descoberta de serviços inter e multi organizacionais, Yale-Loehr et al. (2010) desenvolvem uma abordagem semi-automática para recomendar serviços compartilhados a partir de especificações de requisitos de *software*. Nessa abordagem, é utilizada uma função similaridade semântica para comparar palavras-chaves nas especificações de requisitos por meio dos conjuntos de sinônimos (denominado *synsets*) extraídos da WordNet. Segundo os autores, o algoritmo da função de similaridade compara dois documentos de requisitos e emite um conjunto de requisitos semelhantes. Para isso, o algoritmo recebe como entrada duas listas que representam as palavras-chaves de duas linhas de diferentes documentos de requisitos. Em seguida, cada sinônimo de cada palavra da primeira lista é comparado a cada palavra da segunda lista. Se a semelhança entre as palavras atingir determinado limiar e se o sinônimo estiver contido na outra palavra, elas são consideradas semelhantes. O algoritmo de similaridade retornará um conjunto de palavras semelhantes que será utilizado nas etapas seguintes do sistema de recomendação de serviços compartilhados.

O objetivo do reuso de *software* é reaproveitar partes de um *software* já produzido em novos projetos, visando à diminuição de custos, ao aumento de produtividade e à redução do esforço de desenvolvimento. No trabalho de Bildhauer et al. (2009), os autores apresentam o projeto ReDseeDS que apóia o reuso de partes de *software* baseando-se na verificação de similaridade dos seus requisitos. Segundo os autores, quando se inicia o desenvolvimento de um novo *software*, os novos requisitos são comparados a antigos requisitos armazenados em repositórios de casos passados. A comparação entre os requisitos pode ser feita em três diferentes níveis de granularidade: i) similaridade de palavras básicas; ii) similaridade de sentenças em linguagem natural; e iii) similaridade de cenários. Para calcular a similaridade entre palavras básicas, os autores utilizam a abordagem proposta por Li et al. (2006), em que a medida de similaridade entre palavras é calculada baseando-se no comprimento de caminho mínimo entre dois *synsets* no grafo, já que, na WordNet, os nomes e verbos são arranjados hierarquicamente em um grafo acíclico direcionado pelo relacionamento do tipo *hasHyponym*. Para calcular a similaridade entre as sentenças, os autores extraem, de cada sentença, um vetor de palavras cujo comprimento é o numero de palavras distintas de cada sentença. O valor referente à similaridade de cada palavra do vetor para as demais palavras da sentença é

calculado usando a abordagem proposta por Li et al. (2006). A similaridade entre as sentenças é definida com base no cosseno do ângulo formado entre os vetores.

No trabalho de Gao et al. (2007), os autores apresentam uma metodologia para procurar requisitos para *software* de Planejamento de Recurso Corporativo (do Inglês, *Enterprise Resource Planning*), ou simplesmente ERP. Segundo os autores, os usuários corporativos usam termos diferentes dos usados pelos fornecedores de ERP, e, por isso, é necessário fazer a correspondência entre as atividades empresariais e as funções do sistema ERP. As atividades empresariais e as funções do sistema são representadas no formato de modelo de negócios são construídos com o uso da linguagem VPML. Para medir a similaridade dos modelos, os autores combinam o uso de medidas de similaridade semântica baseada em distância entre conceitos àquelas baseadas em características dos conceitos. A similaridade entre os modelos é verificada em nível de atividade. Uma atividade é definida como um 4-Tupla: (nome, entrada, saída, recurso) em que "nome" refere-se ao identificador da atividade; "entrada" refere-se aos objetos de negócio; "saída" são os formulários ou eventos de saída; e "recurso" refere-se à pessoa ou máquina necessária para a atividade. Para os autores, a verificação de similaridade entre o modelo de atividades fornecido pela empresa e o modelo de referência do *software* ERP ajudará a empresa a decidir por mudar seus processos de negócios ou personalizar o *software* ERP.

Durante o desenvolvimento de um *software*, o analista de requisitos pode pesquisar por funcionalidades relacionadas, ambíguas ou duplicadas. Em Natt och Dag et. al, (2002), os autores apresentam um método para encontrar relacionamentos entre requisitos por meio da análise de similaridade textual com auxílio de medidas estatísticas. Nesse trabalho, os autores utilizam três medidas estatísticas de similaridade para encontrar pares de requisitos textuais duplicados. As medidas de similaridade *Dice*, *Cosine* e Jaccard foram utilizadas para verificar o nível de correspondência entre termos para os campos descrição e resumo dos requisitos analisados. A partir de uma tabela de contingência, os autores avaliaram os resultados obtidos por faixas de limiar de similaridade para uma das medidas. Segundo os autores, o método proposto pode ser eficaz para identificar requisitos duplicados e interdependência entre eles. No entanto, há uma limitação da proposta desse trabalho que está relacionada ao tamanho do texto que compõe os requisitos. Para requisitos descritos com pouco texto, acredita-se que as medidas de similaridade propostas sejam ineficazes, uma vez que, a quantidade de palavras para o levantamento de dados estatísticos é pequena. Nesse caso, o uso de funções de similaridade que considerem o relacionamento entre palavras

distintas, ao invés daquelas que consideram apenas a exata correspondência entre termos, pode minimizar problemas particulares envolvendo similaridade entre textos curtos.

Os trabalhos apresentados nesta seção realizam a comparação dos requisitos por similaridade textual. As estratégias de verificação de similaridade vistas nos trabalhos são baseadas em dados estatísticos, como por exemplo, os dados usados nas funções Dice e Jaccard, em vetores de termos, como a medida *Cosine Similarity* ou baseadas em informações obtidas da WordNet, como a distância entre os conceitos. De modo geral, os trabalhos aqui apresentados utilizam alguma medida de similaridade para identificar características comuns existentes em requisitos.

A abordagem proposta neste trabalho utilizou uma medida de similaridade para comparar estórias de usuário para sugerir possíveis casos de duplicidade e evolução em requisitos. Este trabalho também utilizou uma função de similaridade para apoiar o agrupamento de estórias de usuário e sugerir possíveis alterações em *sprints* do Scrum.

Este capítulo apresentou alguns trabalhos que, de alguma maneira, estão relacionados a abordagem desenvolvida neste trabalho. Foram apresentados trabalhos relacionados a agrupamento de requisitos, a mineração de estórias de usuário e trabalhos que fazem uso de alguma medida de similaridade para analisar requisitos. Os estudos aqui vistos mostram que os requisitos textuais podem ser analisados por uso de agrupamento de textos e por similaridade. No entanto, boa parte destes trabalhos analisam requisitos que possuem mais conteúdo de texto que as estórias de usuário empregadas no Scrum. No geral, as estórias usadas no Scrum são sentenças curtas de texto e, por esta razão, possuem uma quantidade menor de palavras para ser analisada.

6 Metodologia de Pesquisa

Passemos a descrever a metodologia utilizada nos experimentos feitos neste trabalho. Foram realizados experimentos de acordo com os seguintes objetivos: 1) Definição da função de similaridade mais adequada ao problema de comparação de estórias; 2) Identificação de casos de estórias de usuários duplicadas; 3) Identificação de casos de estórias evoluídas; e 4) Análise de distribuição de estórias de usuário em *sprints* de desenvolvimento. Este capítulo está dividido em quatro seções que descrevem a metodologia adotada para cada um dos experimentos realizados.

6.1 Metodologia para o experimento de escolha de função de similaridade

Diante da diversidade de funções de similaridade que podem ser utilizadas para análise de texto fez-se necessária a realização de experimentos que indicassem qual função de similaridade melhor identifica as semelhanças existentes entre estórias de usuário. Dado que uma estória de usuário é constituída basicamente por uma sentença curta de texto, foram definidos dois pressupostos para a seleção da função de similaridade, sendo eles:

Pressuposto 1: Em se tratando da identificação de semelhanças entre sentenças curtas de texto, a função de similaridade mais sensível é aquela que consegue identificar o maior número de pares de texto independentes que, eventualmente, possuam alguma semelhança; e,

Pressuposto 2: Em se tratando da identificação de igualdade entre sentenças curtas de texto, a função de similaridade mais precisa é aquela que produz a maior média de similaridade entre pares de textos semanticamente equivalentes.

Para a seleção da função de similaridade mais sensível à identificação de semelhanças em sentenças curtas de texto, optou-se por verificar a similaridade entre pares de sentenças curtas que não necessariamente possuíssem algum relacionamento semântico predefinido. Nesse caso, cada função de similaridade analisou diferentes combinações de pares de estórias de usuários reais, na forma de sentenças curtas de texto, para identificar a quantidade de pares de estórias que possuíssem alguma semelhança.

Para a seleção da função de similaridade mais precisa à identificação de igualdade em sentenças curtas de texto, cada função calculou a similaridade de um conjunto de pares de sentenças curtas equivalentes, $P_i(B_i, R_i)$, tal que P_i corresponde ao *i*-ésimo par de sentenças, B_i corresponde à *i*-ésima sentença base, e R_i à *i*-ésima sentença reescrita gerada a partir da sentença base B_i, tal que, B_i e R_i são sentenças semanticamente equivalentes. A média aritmética de similaridade obtida após a comparação dos pares de sentenças foi utilizada como parâmetro para a verificação da precisão da função de similaridade. O cálculo da média de similaridade (MS) é descrito na Equação 24.

$$MS = \frac{1}{n}\sum_{i=1}^{n} P_i(B_i, R_i) \tag{24}$$

Para escolher a função de similaridade foram analisadas três funções de similaridade semântica que fazem uso do banco de dados lexical WordNet, uma função de similaridade para uso no Modelo de Espaço Vetorial, e uma função baseada na ocorrência de termos comuns. Essas funções foram selecionadas para que fossem verificadas duas estratégias para a análise de similaridade de textos: a que se baseia na análise semântica das palavras e a que é apoiada por dados estatísticos extraídos por meio da análise do número de ocorrências de palavras.

6.2 Metodologia para o experimento de identificação de estórias duplicadas

O objetivo deste experimento é verificar se é possível identificar casos de estórias de usuários duplicadas por meio da utilização de uma função de similaridade. A hipótese adotada é que uma estória de usuário *n,* pertencente ao conjunto de estórias novas *N,* seja considerada uma versão duplicada da estória original *o*, pertencente ao conjunto de estórias originais *O,* quando *n* e *o* possuírem um nível de similaridade maior ou igual a um determinado limiar informado pelo usuário.

A função de similaridade utilizada neste experimento foi a função escolhida por meio do experimento descrito na seção 6.1. A ferramenta de apoio à decisão desenvolvida

neste trabalho utilizou tal função de similaridade e comparou as estórias de usuários reais e estórias de usuários duplicadas produzidas para este experimento. A ferramenta de apoio à decisão comparou estórias originais com novas estórias, de fato duplicadas, e após analisar um limiar de similaridade informado, indicou os possíveis casos encontrados de estórias duplicadas. O número de pares de estórias de usuário que foram comparados é igual a *num_O x num_N*, tal que *num_O* é o número de estórias originais e *num_N* é o número de novas estórias de usuário duplicadas. No Quadro 5, a seguir, é apresentado o algoritmo de indicação de estórias duplicadas utilizado pela ferramenta de apoio à decisão.

Quadro 5 - Algoritmo para indicação de estórias duplicadas

Algoritmo para indicação de duplicidade de estórias de usuário

entrada: estórias de usuários originais $O = \{o_1, o_2, \ldots, o_n\}$
estórias de usuários novas $N = \{n_1, n_2, \ldots, n_k\}$
limiar de similaridade L

saída: casos de duplicidade $Dup = \{(o_1\ n_1), \ldots, (o_n\ n_k)\}$

para cada $o \in O$ **faça**
  para cada $n \in N$ **faça**
    - calcular a similaridade S entre o e n;
    se $S >= L$ **então**
      - indicar como possível caso de duplicidade o par de estórias o,n;
    fimse
  fimpara
fimpara

Para avaliação dos resultados, foram considerados os números de verdadeiro positivos (VP), falso positivos (FP), verdadeiro negativos (VN) e falso negativos (FN). Um caso verdadeiro positivo é aquele que para o qual uma real estória duplicada é indicada pelo algoritmo; um caso falso positivo é aquele em que uma estória não duplicada é erroneamente indicada pelo algoritmo como sendo uma estória duplicada; um caso verdadeiro negativo ocorre quando uma estória não duplicada é corretamente identificada pelo algoritmo e; um caso falso negativo é aquele par o qual uma real estória duplicada é classificada pelo algoritmo como estória não duplicada. A Tabela 1 apresenta a tabela de contingência com os possíveis resultados para a indicação de duplicações de estórias de usuário.

Tabela 1 - Tabela de contingência com os resultados para indicação de estórias duplicadas

	Abaixo do limiar de similaridade	**Acima ou igual ao limiar de similaridade**	**Total**
Estórias não duplicadas	Verdadeiro Negativo (VN)	Falso Positivo (FP)	VN + FP
Estórias duplicadas	Falso Negativo (FN)	Verdadeiro Positivo (VP)	FN + VP
Total	VN + FN	FP + VP	VN + FP + FN + VP

A partir dos números conseguidos para os casos verdadeiro positivos, falso positivos, verdadeiro negativos e falso negativos foram calculadas as medidas *Recall* (revocação), *Precision* (precisão) e *Accuracy* (acurácia). Com base nas medidas *Recall* e *Precision* foi calculado a medida *F-measure* (medida F). Tais medidas são descritas em Rijsbergen (1979) e são geralmente utilizadas para analisar o desempenho de classificadores, como por exemplo, a tarefa de categorização de textos visto em Yang (1999).

As medidas *recall, precision, F-measure* e *accuracy* são descritas nas Equações 25, 26, 27 e 28 respectivamente. A medida *recall* refere-se a completude ou cobertura e diz o percentual de casos de estórias duplicadas identificadas corretamente dentre todas as estórias realmente duplicadas. A medida *precision* diz o percentual de estórias duplicadas identificadas corretamente dentre todas as que foram classificadas como duplicadas. Já, a medida *F-measure,* equilibra as medidas *recall* e *precision* procurando dar a elas pesos iguais o que permitirá analisar, de forma unificada, o percentual de cobertura de estórias duplicadas e a precisão do método de identificação de duplicidades. Com a medida *F-measure* é possível ter uma avaliação sobre o desempenho do método de identificação de duplicidades com base na cobertura e precisão alcançada pelo método. Por fim, a medida *accuracy* diz quão efetivas são as indicações verdadeiras, ou seja, o percentual de estórias realmente duplicadas e estórias realmente não duplicadas que foram corretamente classificadas.

$$recall = \frac{VP}{VP + FN} \tag{25}$$

$$precision = \frac{VP}{VP + FP} \tag{26}$$

$$F\text{-}measure = 2\ x\ \frac{recall\ x\ precision}{recall + precision} \tag{27}$$

$$accuracy = \frac{VP + VN}{VP + FP + VN + FN} \tag{28}$$

As estórias de usuário duplicadas utilizadas neste experimento foram geradas a partir de estórias reais de três diferentes bases com o uso de ferramentas automáticas de reescrita de texto. As ferramentas utilizadas neste experimento parafrasearam o texto da estória original, gerando uma estória duplicada com texto de grafia distinta, mas com mesmo significado do texto da estória original. A Figura 16 apresenta um esquema em que se apresentam os conjuntos de estórias duplicadas (X_1, X_2 e X_3; Y_1, Y_2 e Y_3; Z_1, Z_2 e Z_3) gerados a partir das ferramentas de reescrita de texto ou parafraseadores (P_1, P_2 e P_3), para as bases de estórias A, B e C. Amostras extraídas dos conjuntos de estórias duplicadas foram utilizadas neste experimento.

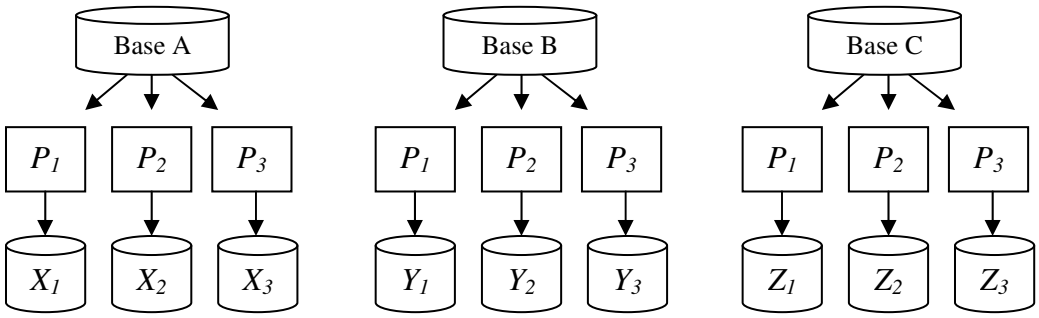


Figura 16 - Esquema de bases de estórias e respectivos conjuntos de estórias duplicadas

As estórias de usuário originais e estórias de usuário duplicadas analisadas neste experimento foram submetidas aos processos de preprocessamento tokenização, lematização e remoção de *stopwords*. Para este experimento, foi considerado apenas o trecho condizente à funcionalidade tanto das estórias de usuário originais, quanto das estórias duplicadas.

O Quadro 6 apresenta alguns exemplos de trechos de funcionalidades de estórias de usuário originais e estórias de usuário duplicadas.

Quadro 6 - Exemplos de funcionalidades de estórias de usuário duplicadas

Funcionalidade Original: *I want to send files to allow students to download* **Funcionalidade Duplicada**: *I need to send documents to permit pupils to download* **Funcionalidade Original**: *I want to send announcements to the students* **Funcionalidade Duplicada**: *I need to send declarations to the pupils* **Funcionalidade Original**: *I want to see messages or warnings given by the teacher to me* **Funcionalidade Duplicada**: *I need to receive communications or notices given by the instructor to me*

Em um cenário real de uso da metodologia ágil Scrum, este experimento indicará se o uso de uma função de similaridade pode indicar a existência de duplicidade de estórias de usuário entre novas estórias de usuário que serão inseridas no *backlog* do produto e estórias de usuário não implementadas.

6.3 Metodologia para o experimento para identificação de estórias evoluídas

O objetivo deste experimento é verificar se estórias de usuário que sofrerão evolução podem ser identificadas por intermédio de uma função de similaridade. A hipótese adotada neste experimento é que uma estória de usuário mantém um nível de relacionamento com suas versões anteriores mesmo após diferentes ciclos de evolução. Portanto, para uma estória de usuário *o*, pertencente ao conjunto de estórias originais *O,* sofrer alteração ou evolução por uma estória de usuário *n,* pertencente ao conjunto de estórias novas *N,* é necessário que *o* e *n* possuam um nível de similaridade maior ou igual a um determinado limiar que será informado pelo usuário.

A função de similaridade utilizada neste experimento será a função escolhida por meio do experimento descrito na seção 6.1. A ferramenta de apoio à decisão desenvolvida neste trabalho utilizou a função de similaridade e comparou estórias de usuários reais e novas estórias de usuários modificadoras produzidas para este experimento. A ferramenta de apoio à decisão comparou estórias de usuário originais com novas estórias de usuário, de fato modificadoras, ou seja, estórias que realizam alterações em outra estória. A partir de um limiar de similaridade informado, a ferramenta indicou os possíveis casos de estórias que sofrerão alteração ou evolução. O Quadro 7 apresenta o algoritmo utilizado pela ferramenta de apoio à decisão para a indicação de estórias de usuário que sofrerão evolução.

Quadro 7 - Algoritmo para indicação de estórias que sofrerão evolução

Algoritmo para indicação de evolução de estórias de usuário

entrada: estórias de usuários originais $O = \{o_1, o_2, \ldots, o_n\}$
estórias de usuários novas $N = \{n_1, n_2, \ldots, n_k\}$
limiar de similaridade L

saída: casos de evolução $E = \{(o_1\ n_1), \ldots, (o_n\ n_k)\}$

para cada $o \in O$ **faça**
  para cada $n \in N$ **faça**
    - calcular a similaridade S entre o e n;
    se $S >= L$ **então**
      - indicar como possível caso de evolução a estória o;
    fimse
  fimpara
fimpara

O número de pares de estórias de usuário que foram comparados é igual a *num_O x num_N*, tal que *num_O* é o número de estórias originais e *num_N* é o número de novas estórias de usuário modificadoras.

De forma similar à metodologia descrita na seção 6.2, os resultados foram avaliados a partir dos números de verdadeiro positivos, falso positivos, verdadeiro negativos e falso negativos. Os resultados possíveis para a indicação de estórias que evoluirão são apresentados na Tabela 2. As medidas *Recall* (revocação), *Precision* (precisão), *F-measure* (medida F) e *Accuracy* (acurácia), apresentadas na seção 6.2, também foram utilizadas para avaliação do experimento.

Tabela 2 - Tabela de contingência com os resultados para indicação de estórias evoluídas

	Abaixo do limiar de similaridade	**Acima ou igual ao limiar de similaridade**	**Total**
Estórias não evoluídas	Verdadeiro Negativo (VN)	Falso Positivo (FP)	VN + FP
Estórias evoluídas	Falso Negativo (FN)	Verdadeiro Positivo (VP)	FN + VP
Total	VN + FN	FP + VP	VN + FP + FN + VP

As estórias de usuário modificadoras utilizadas neste experimento foram produzidas manualmente a partir de estórias de usuário originais escolhidas aleatoriamente. As estórias de usuários modificadoras são versões que se relacionaram com as estórias originais por meio de modificações nas funcionalidades descritas nas estórias originais. Podem ser consideradas modificações na funcionalidade, por exemplo, os acréscimos de detalhes à funcionalidade original, as funcionalidades complementares à funcionalidade original, e a adição de restrições à funcionalidade.

As estórias de usuário originais e estórias de usuário modificadoras utilizadas neste experimento foram submetidas aos processos de preprocessamento tokenização, lematização e remoção de *stopwords*. Para este experimento, foi considerado apenas o trecho condizente à funcionalidade tanto das estórias de usuário originais, quanto para as estórias modificadoras. O Quadro 8 apresenta alguns exemplos de trechos de funcionalidades de estórias de usuário originais e de estórias de usuário modificadoras.

Quadro 8 - Exemplos de funcionalidades de estórias de usuário modificadoras

Tipo de Alteração: acréscimo de detalhes à funcionalidade original **Funcionalidade Original**: *I am notified about the results of surveys about my classes* **Funcionalidade Modificadora**: *I want to receive a PDF file by e-mail with data of surveys about my classes* **Tipo de Alteração**: funcionalidade complementar **Funcionalidade Original**: *I want to generate password to hand out to students* **Funcionalidade Modificadora**: *I want to change the password to students* **Tipo de Alteração**: restrição de funcionalidade **Funcionalidade Original**: *I can edit any site member profile* **Funcionalidade Modificadora**: *I can edit only Practitioner profile*

Em um cenário real de uso da metodologia ágil Scrum, este experimento indicará se o uso de uma função de similaridade pode indicar a existência de casos de evolução de estórias de usuário já implementadas quando da chegada de novas estórias de usuário.

6.4 Metodologia para análise de distribuição de estórias por meio de técnica de agrupamento de textos

Na fase planejamento de *sprint*, a equipe de desenvolvimento Scrum, junto ao dono do produto, realiza a escolha de estórias a serem colocadas em *sprints* de desenvolvimento. O objetivo deste experimento é verificar se o uso da técnica de agrupamento de textos contribui para a análise da distribuição de estórias de usuário em *sprints* feitas na fase de planejamento de *sprint.* Para isso, a ferramenta utilizou um algoritmo de agrupamento particional e comparou os grupos de estórias gerados por esse algoritmo com os grupos de estórias informados como *sprints*.

A hipótese adotada neste experimento é que o resultado do agrupamento pode confirmar a escolha de um conjunto de estórias para determinados *sprints*, ou seja, caso as estórias selecionadas para um *sprint* também apareçam integral ou parcialmente em um mesmo *cluster* de estórias, isso indicará que elas foram selecionadas adequadamente para o *sprint* em questão. Caso um determinado *sprint* contenha, por exemplo, 10 estórias e 5 dessas estórias foram agrupadas em um mesmo grupo por meio do agrupamento de texto, isso apoiaria a escolha dessas 5 estórias para o *sprint*.

Para este experimento, um conjunto de estórias de usuário foi analisado por um analista humano e distribuído por ele em *sprint*s de desenvolvimento. A distribuição de estórias produzida pelo analista humano refere-se a grupos de estórias de usuário que esse analista humano julgou serem aptos para se tornarem *sprint*s. Em seguida, a ferramenta agrupou, de forma automática, as estórias de usuário, por meio de um algoritmo de agrupamento e comparou o resultado produzido com os *sprints* de estórias sugeridos pelo analista humano. A Figura 17 mostra o processo de análise de *sprint*s por meio do agrupamento de estórias.

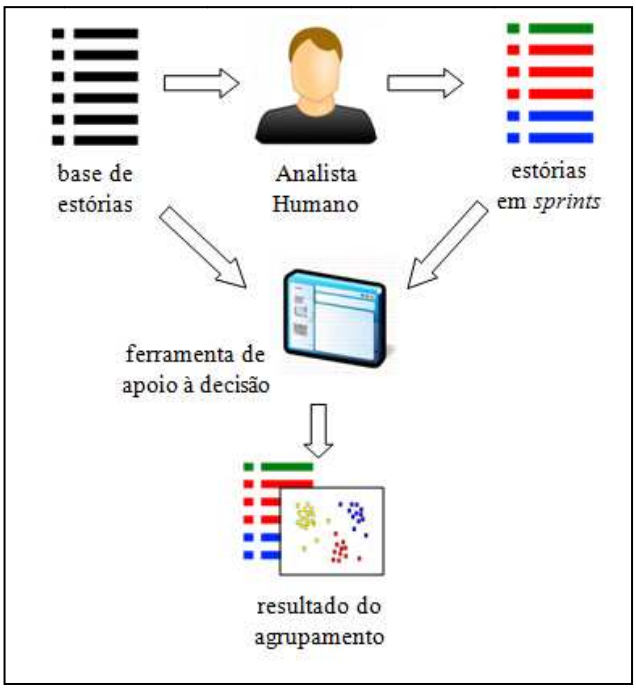


Figura 17 - Processo de análise de *sprint*s por meio do agrupamento de estórias

O processo de agrupamento de estórias de usuário consiste em separar estórias de usuários semelhantes em grupos específicos. O processo de agrupamento utilizou uma função de similaridade para identificar qual grupo seria o mais adequado uma estória de usuário pertencer com base nas semelhanças desta estória com as demais do grupo. Para a análise de semelhança entre as estórias de usuário, foi utilizada a função de similaridade escolhida a partir do experimento descrito na seção 6.1.

O algoritmo *k-medoids* utilizado neste experimento precisa de alguns parâmetros de entrada para ser executado, como, por exemplo, o número de grupos desejados e os pontos iniciais dos grupos para comparação (medoides). O número de grupos desejados foi

informado pelo usuário, enquanto que os medoides iniciais foram selecionados aleatoriamente. A seleção aleatória de medoides reforça a característica de aprendizado não supervisionado do algoritmo e minimiza qualquer influência humana no resultado do agrupamento. Entretanto, a seleção aleatória de valores dos medoides pode produzir resultados de agrupamento discretamente diferentes, mas convergentes.

Os resultados de agrupamento de estórias foram comparados com os *sprint*s de estórias produzidos pelo analista humano. Para cada um dos *sprint*s sugeridos pelo analista, foram computados os grupos de estórias que pertenceriam a tal *sprint* e que também pertenceriam a um determinado *cluster* proposto pelo algoritmo de agrupamento. As estórias atribuídas pelo analista humano a um *sprint* e que também tenham sido atribuídas a um único *cluster* pertencem a uma associação denominada *relacionamento sprint-cluster* (RSC).

O relacionamento *sprint-cluster* entre um *sprint S* e um *cluster C* contêm estórias de usuário relacionadas. Este relacionamento ocorre quando duas ou mais estórias de usuário manualmente atribuídas a *S* por um analista humano também são atribuídas a *C* pelo algoritmo de agrupamento. Formalmente, sendo o *sprint S*={e_1, e_2, e_n} e o *cluster C*={e_1, e_2, e_n}, em que cada e_n corresponde a uma estória de usuário, um relacionamento *sprint-cluster* entre *S* e *C* é dado conforme Equação 29:

$$RSC(S) = S \cap C \mid 2 \leq t(S \cap C) \leq t(S) \tag{29}$$

em que $t(x)$ representa o número total de estórias de usuário de um conjunto.

A Figura 18 descreve uma representação de ocorrência de um *relacionamento sprint-cluster*, em que os pontos vermelhos representam estórias de usuários que são compartilhadas entre o *sprint* e o *cluster*.

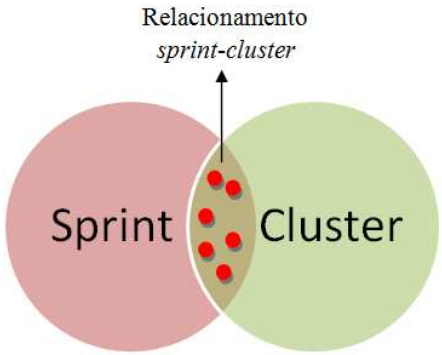


Figura 18 - Exemplo de relacionamento *sprint-cluster*

Em um *sprint* proposto por um analista humano, podem existir mais de uma ocorrência de relacionamento *sprint-cluster*. Nesse caso, são verificados grupos de estórias de usuário semanticamente relacionadas que estão distribuídas por diferentes *clusters*. Esse fenômeno é chamado *relacionamento sprint-cluster distribuído.* A Figura 19 apresenta um exemplo desse tipo de relacionamento.

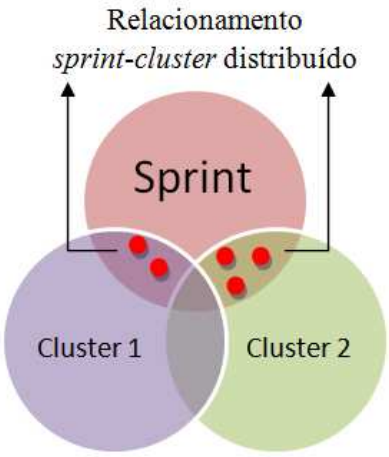


Figura 19 - Exemplo de relacionamento *sprint-cluster* distribuído

Devido ao fato do algoritmo de agrupamento produzir resultados de agrupamento discretamente diferentes, mas convergentes, a cada execução, foram executadas *n* rodadas de execução do algoritmo. Após o término, foi calculada a média aritmética do número de ocorrências de relacionamento *sprint-cluster* para cada *sprint* proposto pelo analista humano. O número médio de relacionamento *sprint-cluster* (M_RSC) é descrito na Equação 30:

$$M\_RSC = \frac{1}{n}\sum_{i=1}^{n} q(RSC(S_i))$$

(30)

em que $q(RSC(S_i))$ representa a quantidade de relacionamentos *sprint-cluster* encontrados para o *i*-ésimo *sprint* S_i, e *n* representa o número de rodadas do algoritmo de agrupamento.

Neste experimento, também foi computada a média aritmética do número de estórias que integravam os relacionamentos *sprint-cluster* encontrados por *sprint*. O número médio de estórias existentes em um relacionamento *sprint-cluster* é calculado conforme a Equação 31:

$$M\_EST = \frac{1}{n}\sum_{i=1}^{n} t(RSC(S_i))$$

(31)

em que $t(RSC(S_i))$ representa o número total de estórias contidas no relacionamento *sprint-cluster* encontrado para o *i*-ésimo *sprint* S_i, e *n* representa o número de rodadas do algoritmo de agrupamento.

As estórias de um *sprint* que não participam de algum relacionamento *sprint-cluster* são aquelas que foram individualmente separadas e colocadas em *clusters* distintos. Formalmente, sendo o *sprint* S={e_1, e_2, e_n} e o *cluster* C={e_1, e_2, e_n}, em que cada e_n corresponde a uma estória de usuário, uma estória de um *cluster* independente (ECI) para o *sprint S* é dado conforme Equação 32:

$$ECI(S) = S \cap C \mid t(S_i \cap C) = 1$$

(32)

em que *t*(*x*) representa o número total de estórias de usuário de um conjunto.

Ainda neste experimento, foi computada a média aritmética do total de estórias que foram individualmente separadas em *clusters* distintos. O número médio de estórias em *clusters* independentes (M_ECI) é calculado conforme Equação 33.

$$M\_ECI = \frac{1}{n}\sum_{i=1}^{n} q(ECI(S_i))$$

(33)

em que $q(ECI(S_i))$ representa a quantidade de estórias em *clusters* independentes encontradas no *sprint* S_i, e n representa o número de rodadas do algoritmo de agrupamento.

Este capítulo apresentou a metodologia utilizada nos experimentos desenvolvidos neste trabalho. Foi descrita a metodologia que orientou a escolha da função de similaridade, em que foram definidos os pressupostos envolvendo a sensibilidade e a precisão de uma função de similaridade. Para a identificação de estórias duplicadas e estórias evoluídas foram propostos algoritmos para a identificação destas situações em um conjunto de estórias. Foram também definidas as medidas *recall*, *precision*, *F-measure* e *accuracy* para avaliar os métodos propostos para identificar duplicidade e para identificar evolução em estórias de usuário. Na metodologia para análise de distribuição de estórias em *sprints* foi descrita a estratégia de uso da técnica de agrupamento empregada. Também foram definidos os conceitos relacionamento *sprint-cluster*, relacionamento *sprint-cluster* distribuído e estórias em *clusters* independentes.

7 Experimentos e Resultados

Neste capítulo, serão apresentados os resultados conseguidos com os seguintes experimentos:

- Experimentos que permitiram escolher a função de similaridade mais adequada para análise de estórias de usuário, o que sustentou o uso desta função de similaridade nos demais experimentos;
- Experimentos relacionados à identificação de estórias de usuários duplicadas e estórias que evoluíram por meio da chegada de outras estórias;
- Experimento demonstrando como a técnica de agrupamento de texto pode auxiliar o planejamento de *sprint*s.

Para os experimentos descritos neste capítulo, foram utilizadas duas bases de estórias de usuário de projetos desenvolvidos por meio da metodologia ágil *Scrum,* bem como uma base de dados de estória que foi construída por meio da análise de requisitos de um sistema já implementado e em funcionamento. Também foi utilizada uma base de sentenças de texto semanticamente equivalentes para apoiar a escolha da função de similaridade para comparação de estórias de usuário. Essas quatro bases são apresentadas a seguir.

7.1 Bases de dados utilizadas

Para os experimentos que serão descritos neste capítulo, foram utilizadas bases de dados compostas por estórias de usuários reais e por sentenças de texto semanticamente equivalentes. Foram consideradas as seguintes bases de dados:

- *Scrum Alliance Website* (*Scrum Alliance*, 2015);
- Repositório Institucional da Universidade de Bath (Cope, 2013);
- Sistema Escolar Online do Colégio Técnico de Campinas (COTUCA, 2015);
- Base de textos curtos Microsoft Research Paraphrase Corpus - MSRP (Dolan & Brockett, 2005).

As estórias de usuário das três bases de estórias e as sentenças da base de textos curtos foram escritas em linguagem natural em Inglês. Das estórias de usuário, foi considerado apenas o trecho condizente à funcionalidade da estória. Não foram considerados os trechos referentes ao tipo de usuário e à razão da estória, ou seja, dada a estória de usuário "*As a help desk operator I want to search for my customers for their first and last names so that customer response times remain short",* considerou-se apenas o trecho "*I want to search for my customers for their first and last names*".

A seguir, uma breve descrição das bases de dados utilizadas.

7.1.1 Base de estórias - *Scrum Alliance Website* (SAWeb)

A base de estórias SAWeb é um conjunto contendo 99 estórias de usuário extraídas de um exemplo de *backlog* do produto escrito para descrever as funcionalidades iniciais para o website do Scrum Alliance. Esse *backlog* do produto lista tudo o que o dono do produto e a equipe Scrum acreditam que deva ser incluído no *software* que está sendo desenvolvido por meio da metodologia ágil Scrum (*Scrum Alliance*, 2015).

De acordo com o *backlog* do produto, o website a ser construído deve conter funcionalidades que permitam sua utilização por usuários de diferentes papéis, por exemplo, administrador do site, editor do site, membro do site, praticante Scrum, instrutor Scrum, visitantes e patrocinadores. O *backlog* do produto original apresenta as 99 estórias de usuário divididas em 13 categorias. As categorias e a quantidade de estórias são descritas na Tabela 3.

Tabela 3 - Categorias do *backlog* do produto do website *Scrum Alliance*

Categorias	**Qtde de Estórias**
Profiles	14
News	6
Courses and Events	16
FAQs	3
Resource	2
Jobs	9
Articles	14
Home Page	10
Ratings	6
What is Scrum?	2
Registry	7
Membership	8
For Trainers Only	2
Total	99

Características da base de estórias de usuário SAWeb:

- **Tema do software**: *Website* para divulgação de artigos, cursos e outros conteúdos
- **Participantes**: administrador, editor, membro do site, visitante, instrutor, *Scrum Master*, dono do produto e praticantes de Scrum
- **Número de estórias**: 99
- **Idioma**: Inglês
- **Tamanho da estória**: 84 caracteres (média 18 palavras)
- **Conteúdo**: tipo de usuário e objetivo

A seguir, um exemplo de uma estória de usuário da base de estórias SAWeb.

"As a site administrator I can read practicing and training applications and approve or reject them"

As estórias de usuário da base SAWeb podem ser conferidas no Apêndice A.

7.1.2 Base de estórias - Repositório de Dados Institucional da Universidade de Bath (RepBath)

A base de estórias RepBath é baseada em um documento contendo 53 estórias de usuário referentes a um repositório de dados institucional para a Universidade de Bath, Inglaterra. Segundo Cope (2013), esse documento define os nossos requisitos como uma universidade e como esse repositório de dados institucional deve funcionar. O propósito do repositório institucional de dados é ser uma ferramenta de arquivamento de longo prazo para os dados de pesquisa da Universidade de Bath.

O âmbito de aplicação do repositório é registrar e conectar os dados de pesquisa da Universidade de Bath a repositórios externos e arquivá-los e, opcionalmente, publicar bases de dados de investigação que, por algum motivo, não possam ser armazenados externamente. Os usuários finais do repositório serão pesquisadores da Universidade de Bath, estudantes de pós-graduação, pesquisadores externos e público com interesse na reutilização de dados de pesquisa publicados pela universidade.

O documento que descreve as 53 estórias de usuário as organiza com base nos usuários do repositório. A Tabela 4 apresenta os usuários descritos nesse documento e o número de estórias para cada usuário.

Tabela 4 - Usuários do repositório institucional de dados da Universidade de Bath

Usuário	**Total de Estórias**
Depositante	22
Reutilizador dos dados	8
Colaborador Externo	3
Gerente da unidade de pesquisa	1
Administrador do arquivo de dados	6
Gerente de informação de pesquisa	5
Serviço de Tecnologia de Informação	4
Desenvolvedor/Mantenedor de serviços	1
Acadêmico Publicador	1
Agência de financiamento	2
Total	53

Características da base de estórias de usuário RepBath:

- **Tema do software**: Repositório de dados
- **Participantes**: depositante, reutilizador dos dados, colaborador externo, gerente da unidade de pesquisa, administrador do arquivo de dados, gerente de informação de pesquisa, serviço de tecnologia de informação, desenvolvedor/mantenedor de serviços, publicador acadêmico e agência de financiamento
- **Número de estórias**: 53
- **Idioma**: Inglês
- **Tamanho da estória**: 156 caracteres (média 28 palavras)
- **Conteúdo**: tipo de usuário, objetivo e razão da funcionalidade

A seguir, um exemplo de uma estória de usuário da base de estórias RepBath:

"As a depositor I want to allow my collaborators privileged access to datasets so that we continue to have a productive relationship"

As estórias de usuário da base RepBath podem ser conferidas no Apêndice A.

7.1.3 Base de estórias - Sistema Escolar Online do Colégio Técnico de Campinas (COTUCA)

A base de estórias COTUCA é um conjunto de 71 estórias de usuários produzidas por engenharia reversa de parte do sistema escolar atualmente em uso. Essa base de estórias descreve funcionalidades de um sistema *online* utilizado por diretores, alunos, professores, coordenadores, bibliotecários e pais ou responsáveis dos alunos. Esse sistema informatizado proporciona serviços como consulta a boletins e históricos escolares, geração de diário escolar, consulta a contratos de estágio, *download* e envio de documentos, envio de e-mail, etc.

A Tabela 5 apresenta os usuários da base de estórias COTUCA e o número de estórias para cada um deles.

Tabela 5 - Usuários do Sistema Escolar *Online* do Colégio Técnico de Campinas

Usuário	Total de Estórias
Diretor	2
Vice-Diretor	3
Coordenador de estágio	8
Secretária	5
Aluno	26
Professor	22
Bibliotecário	2
Pais/Responsável	3
Total	71

Características da base de estórias de usuário COTUCA:

- **Tema do software**: Sistema *online* de administração escolar
- **Participantes**: diretor, vice-diretor, coordenador de estágio, secretária, aluno, professor, bibliotecário, pais e responsáveis
- **Número de estórias**: 71
- **Idioma**: Inglês
- **Tamanho da estória**: 90 caracteres (média 18 palavras)
- **Conteúdo**: tipo de usuário, objetivo e razão da funcionalidade

As estórias de usuário da base COTUCA podem ser conferidas no Apêndice A. Um exemplo de uma estória de usuário da base de estórias COTUCA é mostrado a seguir:

"As a student I want to consult my grades so that I do not go to school"

7.1.4 Base de dados - Microsoft Research Paraphrase Corpus (MSRP)

A base de dados MSRP contém 5801 pares de sentenças curtas de texto. Segundo Dolan e Brockett (2005), esse *corpus* foi criado utilizando técnicas de extração heurística em conjunto com um classificador baseado em máquinas de vetores de suporte ou SVM (do Inglês, *Support Vector Machines*) para selecionar paráfrases em nível de sentença de um

grande banco de dados de notícias agrupadas por tópicos. A base de dados MSRP foi analisada por humanos que confirmaram que 67% dos pares de sentenças são, de fato, semanticamente equivalentes.

A base de dados MSRP é dividida em duas bases de sentenças: i) base de testes contendo 1725 pares de sentenças, e ii) base de treinamento contendo 4076 pares de sentenças. Para o experimento deste trabalho, foram consideradas 2657 pares de sentenças classificadas como equivalentes extraídas da base de treinamento.

Nas seções seguintes, serão apresentados os experimentos para definição de uma função de similaridade, identificação de estórias de usuários duplicadas e estórias evoluídas e um experimento para análise de distribuição de estórias de usuário em *sprints* de desenvolvimento.

7.2 Experimento 1 - Escolha da função de similaridade

Para a realização dos experimentos foram selecionadas e analisadas três funções de similaridade semântica apoiadas na base de dados lexical WordNet, uma função de similaridade para uso no Modelo de Espaço Vetorial (do Inglês, *Vector Space Model*) e uma função de similaridade baseada na ocorrência de termos comuns. As funções de similaridade semântica escolhidas foram WuP (Wu & Palmer, 1994), Lin (Lin, 1998) e Lesk-A (S. Banerjee & Pedersen, 2002). As funções WuP, Lin e Lesk-A foram selecionadas para testar as três abordagens de análise de similaridade com uso da WordNet apresentadas neste trabalho, sendo elas, a abordagem baseada no caminho entre os conceitos, a abordagem baseada no conteúdo da informação e abordagem baseada em definições. As funções de similaridade baseadas no Modelo de Espaço Vetorial e na ocorrência de termos escolhidas para o experimento foram, respectivamente, *Cosine Similarity* (Palmer & Liang, 1992) e Coeficiente de Jaccard (Jaccard, 1901). As funções *Cosine Similarity* e Coeficiente de Jaccard foram selecionadas por se tratarem de funções usualmente empregadas em análise de textos e por terem sido utilizadas em trabalhos similares envolvendo análise de requisitos.

As cinco funções de similaridade foram utilizadas para verificar o nível de similaridade existente entre pares de textos curtos. Os textos curtos utilizados neste experimento foram submetidos aos processos de preprocessamento tokenização, lematização

(no caso das funções WuP, Lin e Lesk-A), estemização (no caso das funções *Cosine Similarity* e Coeficiente de Jaccard) e remoção de *stopwords*. Todas as funções de similaridade utilizadas neste experimento fornecem como resultado um valor que indica a semelhança entre dois textos curtos. O resultado retornado por essas funções é um valor entre 0 e 1, sendo que valores próximos a zero indicam baixa ou nenhuma similaridade semântica e valores próximos a 1 indicam alta ou total similaridade semântica.

O *toolkit* SEMILAR (Rus, Lintean, Banjade, Niraula, & Stefanescu, 2013) foi utilizado para verificar a similaridade semântica em nível de sentença-sentença entre os pares de estórias de usuários. Para a identificação da função de similaridade mais sensível, foram utilizadas as bases de estórias SAWeb, RepBath e COTUCA. Para a identificação da função de similaridade mais precisa, foi utilizada a base de dados MSRP.

A seguir, serão apresentados os resultados dos experimentos feitos para identificar a função de similaridade mais sensível e mais precisa para análise de textos curtos.

7.2.1 Resultado da identificação da função de similaridade mais sensível

Denomina-se aqui como função mais sensível aquela que identificar o maior número de pares de estórias que possuam alguma similaridade.

Para a base de estórias de usuário SAWeb, foram extraídas, do conjunto de 99 estórias de usuário, as médias aritméticas do nível de similaridade de 4851 combinações diferentes de pares de estórias, a partir das funções de similaridade WuP (Wu & Palmer, 1994), Lin (Lin, 1998), Lesk-A (S. Banerjee & Pedersen, 2002), *Cosine Similarity* (Palmer & Liang, 1992) e Coeficiente de Jaccard (Jaccard, 1901). A Tabela 6 apresenta a média de similaridade conseguida para cada uma das funções de similaridade analisadas. A média de similaridade foi obtida a partir do nível de similaridade encontrado entre cada estória de usuário e as demais estórias de usuário restantes da base de dados. A Tabela 6 também apresenta o número de pares de estórias com alguma similaridade, o número de pares de estórias sem qualquer similaridade e o número de pares de estórias com nível de similaridade acima da média (PSAM).

Tabela 6 - Dados extraídos do experimento a partir da base de estórias SAWeb

Base de Estórias SAWeb					
Função	**Média de similaridade**	**Pares com alguma similaridade (*sim* > 0)**	**Pares sem similaridade (*sim* = 0)**	**Pares com similaridade acima da média (PSAM)**	**PSAM %**
WuP	0,169613	3901	950	2133	43,9
Lin	0,091062	2716	2135	1840	37,9
Lesk-A	0,005168	104	4747	104	2,1
Cosine	0,024686	682	4169	682	14,1
Jaccard	0,023717	682	4169	682	14,1

A Tabela 6 mostra que a função de similaridade semântica WuP (Wu & Palmer, 1994) apresentou a maior média de similaridade comparando-se cada estória de usuário às demais estórias restantes. Observa-se, na Tabela 6, que a função WuP foi a que identificou o maior número de pares de estórias com alguma semelhança, apresentando 3901 pares de estórias com tal característica. Observa-se também que a função WuP foi a que menos indicou pares de estória totalmente distintas (nesse caso, 950 pares de estórias). Os resultados de similaridade produzidos pelas funções em estudo indicaram que a função WuP apresentou similaridade acima da média para 43,9% das 4851 combinações diferentes de estórias de usuário. A função WuP indicou 2133 pares de estórias com índice de similaridade acima da média, seguida da função Lin, que indicou 1840 pares de estórias; da função Lesk-A, que indicou 104 pares de estórias; e *Cosine* e Jaccard ambas indicando 682 pares de estórias.

Para a base de estórias de usuários RepBath, foram extraídas, do conjunto de 53 estórias de usuário, as médias aritméticas do nível de similaridade de 1378 combinações diferentes de pares de estórias, a partir das funções de similaridade WuP, Lin, Lesk-A, *Cosine Similarity* e Coeficiente de Jaccard (Jaccard, 1901). A Tabela 7 apresenta as médias de similaridade conseguidas comparando-se cada par de estórias de usuário, o número de pares de estória com alguma similaridade, o número de pares de estórias sem qualquer similaridade e o número de pares de estórias com nível de similaridade acima da média (PSAM).

Tabela 7 - Dados extraídos do experimento a partir da base de estórias RepBath

Base de Estórias RepBath					
Função	**Média de similaridade**	**Pares com alguma similaridade (*sim* > 0)**	**Pares sem similaridade (*sim* = 0)**	**Pares com similaridade acima da média (PSAM)**	**PSAM %**
WuP	0,163704	1042	336	641	46,5
Lin	0,111257	786	592	540	39,2
Lesk-A	0,013058	77	1301	77	5,6
Cosine	0,022919	398	980	365	26,5
Jaccard	0,039769	398	980	398	28,9

A Tabela 7 mostra que a função de similaridade semântica WuP foi a que apresentou a maior média de similaridade, identificando, em 1042 pares de estórias, algum nível de similaridade, e apresentando similaridade acima da média para 46,5% das 1378 combinações diferentes de estórias de usuário. A função WuP indicou 641 pares de estórias com índice de similaridade acima da média, seguida da função Lin, que indicou 540 pares de estórias; da função Lesk-A, que indicou 77 pares de estórias; e das funções *Cosine* e Jaccard indicando 365 e 398 pares de estórias, respectivamente.

Para a base de estórias de usuário COTUCA, foram extraídas do conjunto de 71 estórias de usuário as médias aritméticas do nível de similaridade de 2485 combinações diferentes de pares de estórias, a partir das funções de similaridade WuP, Lin, Lesk-A, *Cosine Similarity* e Coeficiente de Jaccard (Jaccard, 1901). A Tabela 8 apresenta as médias de similaridade conseguidas comparando-se os pares de estórias de usuário, o número de pares de estória com alguma similaridade, o número de pares de estórias sem qualquer similaridade e o número de pares de estórias com similaridade acima da média (PSAM).

Tabela 8 - Dados extraídos do experimento a partir da base de estórias COTUCA

Base de Estórias COTUCA					
Função	**Média de similaridade**	**Pares com alguma similaridade (*sim* > 0)**	**Pares sem similaridade (*sim* = 0)**	**Pares com similaridade acima da média (PSAM)**	**PSAM %**
WuP	0,176617	1693	792	1090	43,9
Lin	0,089439	921	1564	805	32,4
Lesk-A	0,002759	19	2466	19	0,8
Cosine	0,038005	483	2002	445	17,9
Jaccard	0,039262	483	2002	483	19,4

Os resultados apresentados na Tabela 8 demonstram que a função de similaridade semântica WuP foi a que apresentou a maior média de similaridade, identificando 1693 pares de estórias que possuem alguma similaridade. Observa-se, na Tabela 8, que a função WuP apresentou similaridade acima da média para 43,9% das 2485 combinações diferentes de estórias de usuário. A função WuP indicou 1090 pares de estórias com índice de similaridade acima da média, seguida da função Lin, que indicou 805 pares de estórias; da função Lesk-A, que indicou 19 pares de estórias, e das funções *Cosine* e Jaccard, indicando 445 e 383 pares de estórias, respectivamente.

A Figura 20 apresenta um gráfico demonstrando o número de pares de estórias de usuário com alguma similaridade encontrados nas bases de estórias SAWeb, RepBath e COTUCA utilizando as cinco funções de similaridade em estudo. Nota-se, na Figura 20, que, das funções de similaridade semântica, as funções de similaridade WuP e Lin foram as que apresentaram melhores resultados para as três bases de estórias de usuário analisadas. Nota-se também que, apesar da função de similaridade Lesk-A fazer uso da base lexical WordNet, ela pouco identificou pares de estórias com alguma similaridade.

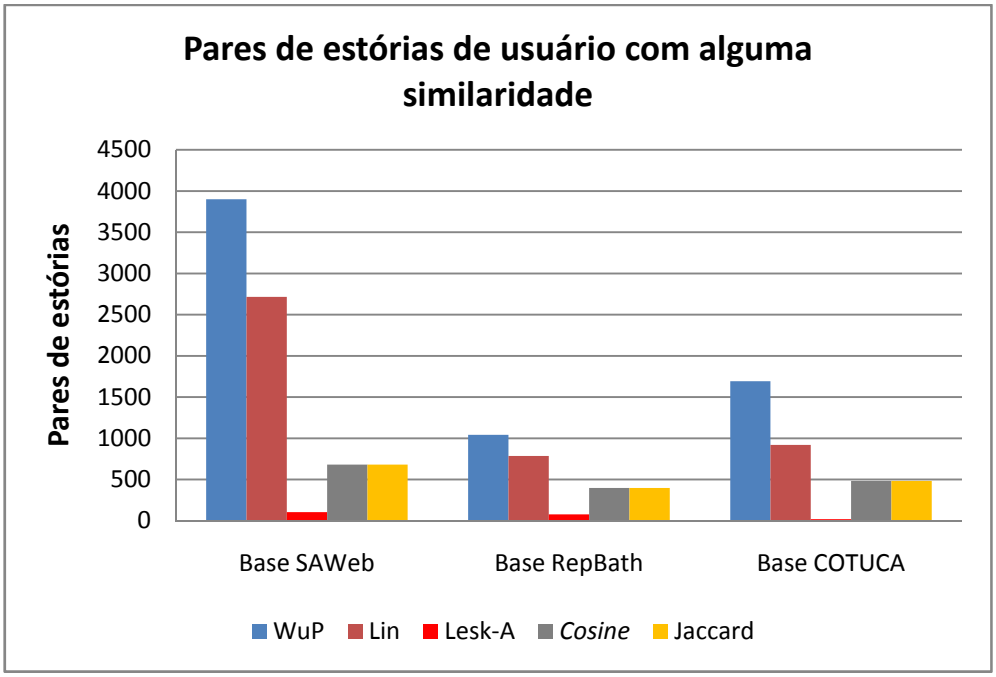


Figura 20 - Gráfico de pares de estórias de usuário com alguma similaridade

Foram analisadas cinco funções de similaridade, comparando-se o nível de similaridade produzido por cada uma delas para diferentes pares de estórias de usuário. Identificou-se que as funções de similaridade semântica WuP e Lin apresentaram resultados melhores que as funções *Cosine* e Jaccard, sendo que estas últimas apresentaram um comportamento similar para as bases de estórias estudadas. Uma das explicações para a superioridade de tais funções é que esse tipo de função utiliza uma fonte adicional de consulta, que é a base de dados lexical WordNet. Nesse caso, as funções de similaridade semântica consideram palavras sinônimas, melhorando significativamente a identificação de relações semânticas, e, consequentemente, a detecção de semelhança entre as estórias de usuário. As funções *Cosine* e Jaccard consideram apenas a ocorrência de palavras exatamente iguais na contabilização de frequência de palavras nas estórias de usuário. Outra explicação para os baixos índices de similaridade encontrados pelas funções *Cosine* e Jaccard deve-se ao fato das estórias de usuário geralmente serem textos curtos e independentes, o que minimiza a ocorrência de palavras comuns entre as estórias, fator importante para o cálculo do escore indicativo do grau de similaridade para essas funções.

Acredita-se que a razão para a baixa sensibilidade da função Lesk-A deve-se ao fato dessa função ser baseada em correspondências entre conceitos. Em se tratando de estórias pouco relacionadas, a comparação por conceitos pode apresentar baixa similaridade, uma vez

que, frases pouco relacionadas tendem a apresentar um número maior de termos diferentes e, portanto, conceitos distintos. Nesse caso, a matriz de similaridade gerada com os pares de estórias por meio da função Lesk-A tende a ser uma matriz esparsa.

Diante das análises realizadas dentre as funções estudadas, a função de similaridade semântica WuP (Wu & Palmer, 1994) foi considerada a mais sensível para se analisar pares de textos curtos, e, consequentemente, estórias de usuários.

7.2.2 Resultado da identificação da função de similaridade mais precisa

Denomina-se aqui como função mais precisa dentre as que estão sendo analisadas aquela que produzir a maior média aritmética de similaridade quando se compara um conjunto de pares de textos curtos predefinidos como equivalentes.

Foram considerados 2657 pares de textos curtos equivalentes extraídos da base de dados MSRP. Foram verificados os níveis de similaridade para os pares de texto equivalentes utilizando as funções de similaridade WuP (Wu & Palmer, 1994), Lin (Lin, 1998), Lesk-A (S. Banerjee & Pedersen, 2002), *Cosine Similarity* (Palmer & Liang, 1992) e Coeficiente de Jaccard (Jaccard, 1901). A Tabela 9 apresenta as médias de similaridade conseguidas comparando os 2657 pares de textos curtos da base de dados MSRP de acordo com a Equação 24 descrita na seção 6.1. Ela também apresenta o número de pares de estórias com similaridade acima da média (PSAM).

Tabela 9 - Médias de similaridade dos pares de textos curtos da base de dados MSRP

Base de dados MSRP			
Função	**Média de Similaridade**	**Pares com similaridade acima da média (PSAM)**	**PSAM %**
WuP	0,726484	1462	55,0%
Lin	0,716533	1421	53,4%
Lesk-A	0,714609	1409	53,0%
Cosine	0,626213	1416	53,2%
Jaccard	0,511197	1274	47,9%

Os resultados apresentados na Tabela 9 demonstram que a função de similaridade semântica WuP foi a que apresentou a maior média de similaridade e a que apresentou o maior número de pares de textos curtos com similaridade acima da média. A Figura 21 apresenta um gráfico em que é possível comparar a média de similaridade encontrada por cada função de similaridade ao se comparar os 2657 pares de textos curtos da base de dados MSRP. Percebe-se que as funções de similaridade semântica WuP, Lin e Lesk-A apresentaram resultados parecidos para a média de similaridade. Os resultados apresentados por essas funções são superiores às médias de similaridade obtidas pelas funções *Cosine* e Jaccard.

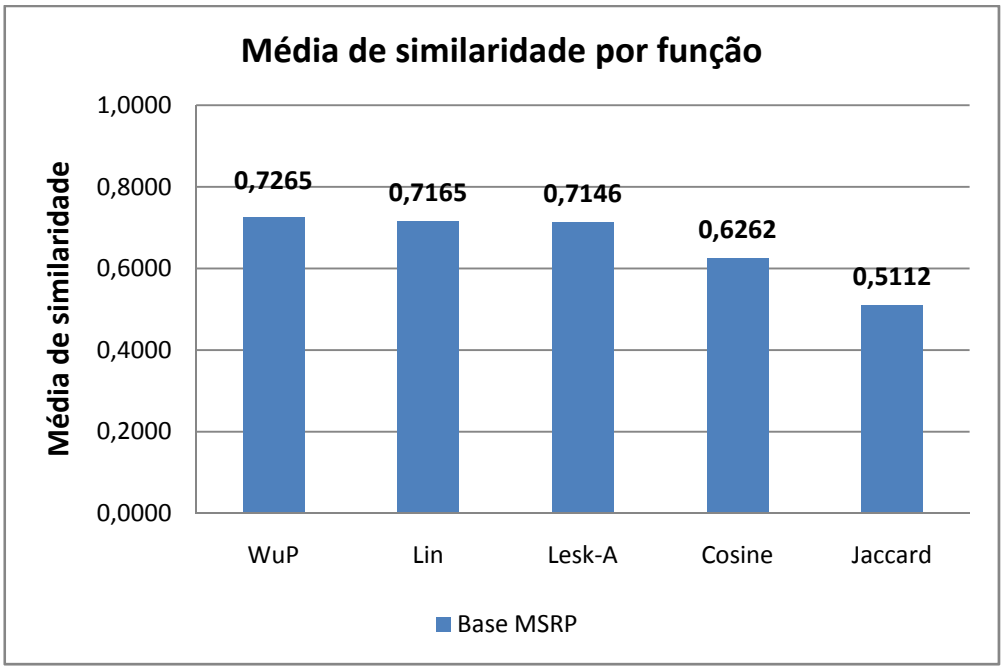


Figura 21 - Gráfico apresentando média de similaridade por função

Diante das análises realizadas dentre as funções estudadas, a função de similaridade semântica WuP (Wu & Palmer, 1994) foi considerada a mais precisa para analisar pares de textos curtos, e, consequentemente, estórias de usuários.

De acordo com os experimentos realizados, a função de similaridade WuP foi apontada como a função mais sensível para a identificação de semelhanças e a mais precisa para a identificação de igualdade em sentenças curtas de texto. Em razão disso, a função WuP foi considerada a mais apropriada para analisar estórias de usuário e foi utilizada para os demais experimentos.

7.3 Experimento 2 - Identificação de estórias duplicadas

Para a identificação de duplicidades de estória de usuário, foram produzidas versões duplicadas de estórias de usuário para as três bases de estórias: SAWeb, RepBath e COTUCA. Para cada uma das bases de estórias, foram criados três conjuntos contendo versões duplicadas de estórias de usuário geradas a partir de três ferramentas distintas de reescrita de texto.

As ferramentas de reescrita de texto utilizadas neste experimento foram *Paraphasing Tool* (Paraphrasing Tool, 2015), *Free Article Spinner* (Free Article Spinner, 2016) e *Article Rewriter Tool* (Article Rewriter Tool, 2016). O Quadro 9 apresenta alguns exemplos de estórias duplicadas geradas para a base de estórias de usuário SAWeb por meio da ferramenta *Paraphasing Tool*. As estórias duplicadas geradas para este experimento referentes a cada base de dados podem ser consultadas no Apêndice B.

Quadro 9 - Exemplos de estórias duplicadas geradas para a base SAWeb

Estória Original: As *a Practitioner I want my profile page to include additional details about me* **Estória Duplicada Gerada**: *As a site member I want to put information about me in my page*
Estória Original: *As a site member I can scroll through a listing of jobs* **Estória Duplicada Gerada**: *As a Practitioner I want to see a job report*
Estória Original: *As a site editor I have pretty good control over how the article looks* **Estória Duplicada Gerada**: *As a site admin I want to control the appearance of the article*

Para cada conjunto de versões de estórias duplicadas, foram selecionadas, aleatoriamente, 30% da quantidade de estórias duplicadas existentes, formando os conjuntos de amostras de duplicidades. Os conjuntos de amostras extraídos foram denominados *SAWeb_Dupl_A*, *SAWeb_Dupl_B* e *SAWeb_Dupl_C* (para a base SAWeb); *RepBath_Dupl_A*, *RepBath_Dupl_B* e *RepBath_Dupl_C* (para a base RepBath); e *COTUCA_Dupl_A*, *COTUCA_Dupl_B* e *COTUCA_Dupl_C* (para a base COTUCA). As amostras de estórias duplicadas foram utilizadas para realizar a comparação com as estórias de usuário originais, ou seja, as estórias das bases SAWeb, RepBath e COTUCA.

Por não ter sido encontrado, na literatura, um trabalho que apresentasse um número médio de duplicações em requisitos de *software*, optou-se por selecionar uma amostra

de 30% das estórias para o experimento, por acreditar-se que o número de duplicações em um ambiente real de desenvolvimento seja próximo desse percentual. A Tabela 10 apresenta o número amostras de estórias duplicadas e aleatoriamente selecionadas para cada uma das base de dados em estudo.

Tabela 10 - Número de estórias duplicadas selecionadas

Base de estórias	**Número de estórias**	**Estórias duplicadas selecionadas**
SAWeb	99	30
RepBath	53	16
COTUCA	71	22

As estórias de usuário originais e as amostras de estórias de usuário duplicadas foram analisadas por meio da ferramenta de apoio à decisão desenvolvida neste trabalho. Essa ferramenta utilizou a função de similaridade WuP (Wu & Palmer, 1994) para sugerir casos de estórias duplicadas com base na análise de similaridade semântica existente entre as estórias de usuário. A função de similaridade foi utilizada para verificar a similaridade semântica apenas do trecho condizente à funcionalidade, tanto das estórias originais, quanto das estórias duplicadas. Os trechos da estória que condizem com o tipo do usuário e a razão da funcionalidade não foram considerados durante a análise para evitar qualquer influência nos resultados de similaridade, uma vez que o conteúdo desses trechos pode aparecer em outras estórias de usuários.

Para a indicação de possíveis casos de duplicidade entre as estórias de usuário foi definido um limiar de similaridade de 50%. Nesse caso, a ferramenta sugerirá um possível caso de duplicidade entre estórias, caso a funcionalidade de uma estória duplicada apresente um valor de similaridade maior ou igual a 50% com qualquer funcionalidade de estória de usuário original. O valor de 50% para o limiar de similaridade foi escolhido por se tratar de um limiar que evitará que os resultados do experimento sejam influenciados positiva ou negativamente. Este valor evitará, por exemplo, que muitos casos de duplicidade sejam sugeridos pela ferramenta - o que significará uma alta taxa de cobertura - mediante a escolha de um baixo valor para o limiar de similaridade. O citado limiar foi adotado para a verificação de duplicidade entre estórias para as bases de estórias SAWeb, RepBath e COTUCA. Um

trabalho que mostra o uso de diferentes valores de limiares de similaridade para a sugestão de casos de duplicidade em estórias de usuário pode ser visto em Barbosa et. al. (2016).

A Tabela 11 apresenta o número de casos de duplicidade sugeridos pela ferramenta e os números de casos verdadeiro positivos, falso positivos, verdadeiro negativos e falso negativos para a base estórias de usuário SAWeb para um limiar de 50% de similaridade utilizando os três conjuntos de amostras de estórias de usuário duplicadas.

Tabela 11 - Resultado da identificação de estórias duplicadas utilizando a base de estórias SAWeb com limiar de similaridade em 50%

Conjuntos de estórias duplicadas	**Número de estórias duplicadas (DUP)**	**Casos de duplicidades sugeridos (SUG)**	**Casos Verdadeiro Positivos (VP)**	**Casos Falso Positivos (FP)**	**Casos Verdadeiro Negativos (VN)**	**Casos Falso Negativos (FN)**
SAWeb_Dupl_A	30	50	18	32	2908	12
SAWeb_Dupl_B	30	58	20	38	2902	10
SAWeb_Dupl_C	30	43	22	21	2919	8
Médias		50,3	20	30,3	2909,7	10

A Tabela 12 apresenta os resultados obtidos para *Recall*, *Precision, F-measure* e *Accuracy* com base nos valores apresentados na Tabela 11.

Tabela 12 - *Recall*, *Precision*, *F-measure* e *Accuracy* obtidos para a base SAWeb

Conjuntos de estórias duplicadas	***Recall* (%)** **VP / (VP+FN)**	***Precision* (%)** **VP / (VP+FP)**	***F-measure* (%)** **2 x (*Prec* x *Recall*) / (*Prec* + *Recall*)**	***Accuracy* (%)** **(VP + VN) / (VP+FP+VN+FN)**
SAWeb_Dupl_A	60,0%	36,0%	45,0%	98,5%
SAWeb_Dupl_B	66,7%	34,5%	45,5%	98,4%
SAWeb_Dupl_C	73,3%	51,2%	60,3%	99,0%
Médias	66,7%	40,6%	50,2%	98,6%

Para a base de estórias de usuário SAWeb, observa-se nas Tabelas 11 e 12 que, apesar do considerável número médio de casos falso positivos, no caso 30,3, o percentual médio de casos reais de duplicidades identificadas dentre todas as duplicidades realmente existentes (*recall*) foi de 66,7%. Para a mesma base de estórias, o percentual de indicações corretas de estórias duplicadas (*precision*) foi de 40,6%. A média de valores para *F-measure* obtidos foi de 50,2%, indicando que, no geral, o desempenho do método de identificação de duplicidades foi em torno de 50%. A média de acurácia desse experimento para a base SAWeb foi de 98,6%, em consequência do número de casos verdadeiro negativos identificados corretamente, nesse caso, algo em torno de 2909 casos.

A Tabela 13 apresenta os números de casos de duplicidades sugeridos e os números de verdadeiro positivos, falso positivos, verdadeiro negativos e falso negativos para um limiar de 50% de similaridade utilizando os três conjuntos de amostras de estórias de usuário duplicadas para a base de estórias RepBath. A Tabela 14 apresenta os resultados para *Recall*, *Precision, F-measure* e *Accuracy* segundo os valores apresentados na Tabela 13.

Tabela 13 - Resultado da identificação de estórias duplicadas utilizando a base de estórias RepBath com limiar de similaridade em 50%

Conjuntos de estórias duplicadas	**Número de estórias duplicadas (DUP)**	**Casos de duplicidades sugeridos (SUG)**	**Casos Verdadeiro Positivos (VP)**	**Casos Falso Positivos (FP)**	**Casos Verdadeiro Negativos (VN)**	**Casos Falso Negativos (FN)**
RepBath_Dupl_A	16	14	11	3	829	5
RepBath_Dupl_B	16	10	8	2	830	8
RepBath_Dupl_C	16	13	10	3	829	6
Médias		12,3	9,7	2,7	829,3	6,3

Tabela 14 - *Recall*, *Precision*, *F-measure* e *Accuracy* obtidos para a base RepBath

Conjuntos de estórias duplicadas	***Recall* (%)** **VP / (VP+FN)**	***Precision* (%)** **VP / (VP+FP)**	***F-measure* (%)** **2** x (***Prec*** x ***Recall***) **/** (***Prec* + *Recall***)	***Accuracy* (%)** **(VP + VN) / (VP+FP+VN+FN)**
RepBath_Dupl_A	68,8%	78,6%	73,4%	99,1%
RepBath_Dupl_B	50,0%	80,0%	61,5%	98,8%
RepBath_Dupl_C	62,5%	76,9%	69,0%	98,9%
Médias	60,4%	78,5%	68,0%	98,9%

Observa-se, na Tabela 14, que o percentual médio de indicações corretas de estórias duplicadas dentre todas as indicações de duplicidade (*precision*) foi de 78,5%, resultado do baixo número médio de casos falso positivos; nesse caso, 2,7 indicações incorretas em média, conforme Tabela 13. Nota-se ainda na Tabela 14, que percentual médio de casos reais de duplicidades identificadas (*recall*) foi de 60,4%. Segundo a média de valores para *F-measure,* em geral, o desempenho do método de identificação de duplicidades foi de 68,0%. A média de acurácia obtida por esse experimento para a base de estórias RepBath foi 98,9%, em consequência do significativo número de casos verdadeiro negativos acertadamente identificados.

A Tabela 15 apresenta o número de casos de duplicidade sugeridos pela ferramenta e os números de casos verdadeiro positivos, falso positivos, verdadeiro negativos e falso negativos para a base estórias COTUCA para um limiar de 50% de similaridade utilizando os três conjuntos de amostras de estórias de usuário duplicadas.

Tabela 15 - Resultado da identificação de estórias duplicadas utilizando a base de estórias COTUCA com limiar de similaridade em 50%

Conjuntos de estórias duplicadas	**Número de estórias duplicadas (DUP)**	**Casos de duplicidades sugeridos (SUG)**	**Casos Verdadeiro Positivos (VP)**	**Casos Falso Positivos (FP)**	**Casos Verdadeiro Negativos (VN)**	**Casos Falso Negativos (FN)**
COTUCA_Dupl_A	22	68	12	56	1484	10
COTUCA_Dupl_B	22	35	13	22	1518	9
COTUCA_Dupl_C	22	30	10	20	1520	12
Médias		44,3	11,7	32,7	1507,3	10,3

A Tabela 16 apresenta os resultados obtidos para *Recall*, *Precision, F-measure* e *Accuracy* com base nos números apresentados na Tabela 15.

Tabela 16 - *Accuracy*, *Recall*, *Precision* e *F-measure* obtidos para a base COTUCA

Conjuntos de estórias duplicadas	***Recall* (%)** **VP / (VP+FN)**	***Precision* (%)** **VP / (VP+FP)**	***F-measure* (%)** **2** x (***Prec*** x ***Recall***) / (***Prec*** + ***Recall***)	***Accuracy* (%)** **(VP + VN) / (VP+FP+VN+FN)**
COTUCA_Dupl_A	54,5%	17,6%	26,6%	95,8%
COTUCA_Dupl_B	59,1%	37,1%	45,6%	98,0%
COTUCA_Dupl_C	45,5%	33,3%	38,5%	98,0%
Médias	53,0%	29,3%	36,9%	97,2%

Observa-se na Tabela 16 que, o experimento utilizando a base de estórias COTUCA revelou o percentual médio de duplicidades reais identificadas de 53%. Esse experimento mostrou um número considerável de casos falso positivos, em média 32,7 casos de indicações incorretas, conforme Tabela 15. Para a base de estórias COTUCA, o

experimento ainda indicou o percentual médio de 29,3% de indicações corretas de estórias duplicadas, o desempenho médio de 36,9%, segundo a média de valores de *F-measure,* e média de acurácia de 97,2%.

A Figura 22 apresenta um gráfico que demonstra o número médio de indicações verdadeiro e falso positivos no que se refere a indicações de estórias duplicadas, para as bases de estórias SAWeb, RepBath e COTUCA, utilizando o limiar de similaridade no valor de 50%. Por meio da Figura 22, é possível observar o número indicações corretas de estórias realmente duplicadas e o número de indicações equivocadas de estórias como sendo duplicadas. Percebe-se que o uso desse valor de limiar de similaridade no valor de 50% promove um considerável número de casos falso positivos para as bases de estórias SAWeb e COTUCA. Entretanto, esse mesmo limiar apresentou um baixo número de indicações falso positivos para a base de estórias RepBath.

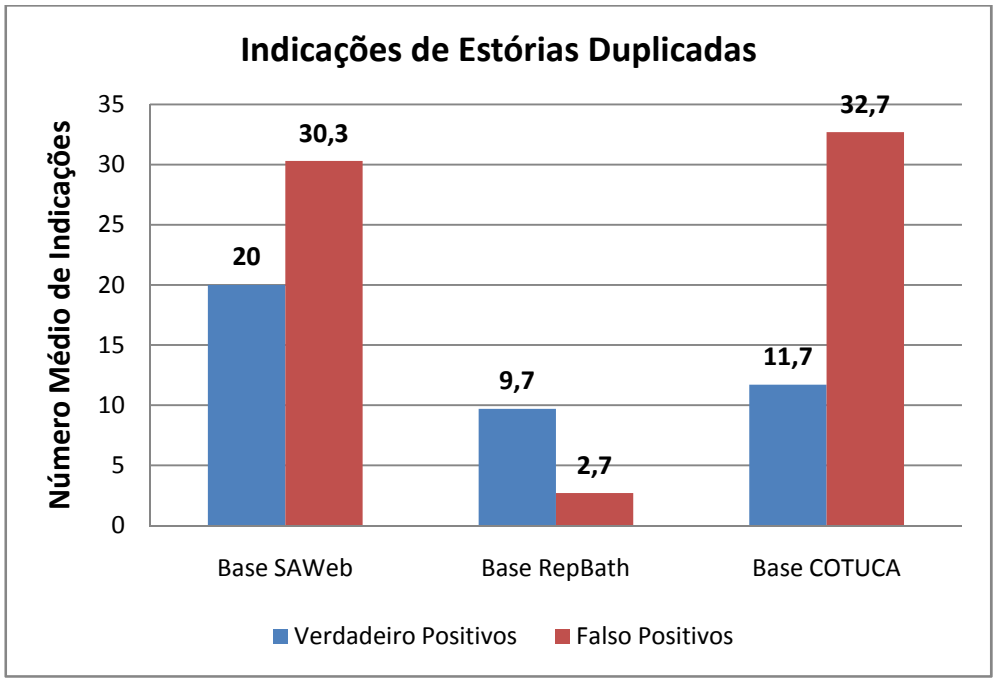


Figura 22 - Gráfico de indicações de estórias duplicadas

A Figura 23 exibe um gráfico que demonstra o valor médio de *recall*, *precision* e *accuracy* para cada uma das bases de estórias analisadas. Para as bases SAWeb e COTUCA, o valor médio para *recall* foi maior que o valor médio de *precision* demonstrando que houve uma maior cobertura de identificação de estórias duplicadas dentre todas as estórias

verdadeiramente duplicadas. Nota-se que a base RepBath apresentou maior valor médio para *precision* demonstrando que, para tal base, as estórias duplicadas e não duplicadas foram melhor identificadas. Além disso, de acordo com os valores obtidos com a medida *F-measure*, o experimento realizado com a base RepBath foi o que apresentou, em geral, o melhor desempenho na identificação de estórias duplicadas. Destacam-se na Figura 23, os valores de acurácia que decorrem do significativo número de casos verdadeiro negativos identificados para as três bases de estória em estudo.

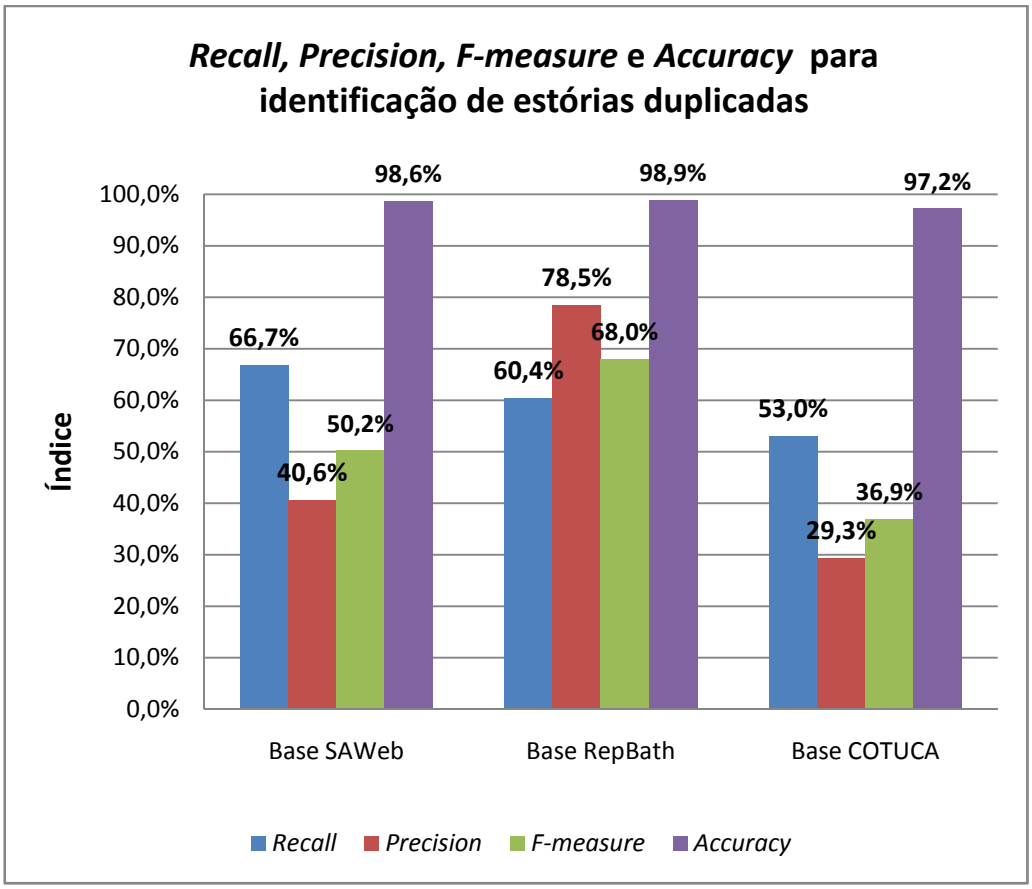


Figura 23 - Médias de *Recall*, *Precision, F-measure* e *Accuracy* para estórias duplicadas

Acredita-se que, em um cenário real de uso da metodologia ágil Scrum, a cobertura (*recall*) da identificação de estórias duplicadas tenha prioridade sobre a precisão, visto que, a não identificação de duplicidades ocasionará a adição desnecessária de uma funcionalidade ao *backlog* do produto deixando-o volumoso e mais difícil de mantê-lo. Além disso haverá o risco de iniciar um trabalho sobre uma funcionalidade que já foi desenvolvida. Um baixo nível de precisão pode fazer com o dono do produto analise casos incorretos de

duplicidade. Apesar desta situação não ser desejada, o custo do trabalho de descarte de casos falso positivos de duplicidade é menor do que o custo gerado pela implementação desnecessária de uma funcionalidade ou o custo de se manter um *backlog* do produto volumoso.

7.4 Experimento 3 - Identificação de estórias que evoluirão

Para a identificação de estórias de usuário que sofrerão evolução foram geradas estórias modificadoras para as três bases de estórias: SAWeb, RepBath e COTUCA. Para cada uma das bases de estórias, foi gerada, manualmente, uma quantia de estórias modificadoras referente a 15% da quantidade de estórias existentes em cada base de dados. Por não ter sido encontrado, na literatura, um trabalho que apresentasse um número médio de estórias que sofrem alterações quando novas estórias de usuário são adicionadas ao *backlog* do produto, considerou-se razoável a parcela de 15% das estórias de um projeto sofrer algum tipo de alteração ou evolução quando da chegada de novas estórias de usuário. A Tabela 17 apresenta o total de estórias modificadoras geradas para cada base de estórias em estudo.

Tabela 17 - Total de estórias modificadoras geradas

Bases de estórias	Número de estórias	Total de estórias modificadoras
SAWeb	99	15
RepBath	53	8
COTUCA	71	11

O Quadro 10 apresenta alguns exemplos de estórias modificadoras que foram manualmente geradas para o experimento envolvendo a base de estórias RepBath. Todas as estórias modificadoras geradas para este experimento referentes a cada base de estórias em estudo podem ser consultadas no Apêndice C.

Quadro 10 - Exemplos de estórias modificadoras geradas para a base RepBath

Estória Original: *As a depositor I want to track downloads of my data so that I can demonstrate the impact of my work* **Estória Modificadora**: *As a depositor I want to see a list of downloads of my data so that I can demonstrate the impact of my work*
Estória Original: *As a depositor I want to deposit the files that I have so that I don't have to spend a lot of time finding the right version and converting to the right format* **Estória Modificadora**: *As a depositor I want to share the file that I deposited so that others have access to materials produced by me*
Estória Original: *As a depositor I want to manage multiple versions of the same dataset so that changes to the dataset are transparent and do not compromise research integrity* **Estória Modificadora**: *As a depositor I want to manage only the last ten versions of a dataset so that Changes to the dataset are transparent and do not compromise research integrity*

As estórias modificadoras foram verificadas com a ferramenta de apoio à decisão desenvolvida neste trabalho. Cada conjunto de estórias modificadoras gerado foi verificado com sua respectiva base de estórias originais. A função de similaridade WuP (Wu & Palmer, 1994) foi utilizada para identificar potenciais casos de estórias originais que serão modificadas por meio de alguma estória modificadora gerada neste experimento. Assim, como no experimento anterior, a função de similaridade foi utilizada para verificar a similaridade semântica existente apenas do trecho da estória de usuário referente à funcionalidade.

Para a indicação de casos de estórias que evoluirão, foi definido um limiar de similaridade de 50%. A ferramenta indicará possíveis casos de estórias que evoluirão quando a funcionalidade de uma estória modificadora apresente um valor de similaridade maior ou igual a 50% em relação a qualquer funcionalidade de estória de usuário original. Como no experimento 2 (seção 7.3), o valor de 50% para o limiar de similaridade foi escolhido por se tratar de um ponto de referência que prevenirá que os resultados do experimento sofram alguma tendência. Esse limiar foi adotado para indicação de evolução de estórias para as bases SAWeb, RepBath e COTUCA.

A Tabela 18 apresenta o número de casos de evolução de estórias sugeridos pela ferramenta e os números de casos verdadeiro positivos, falso positivos, verdadeiro negativos e falso negativos para as bases de estórias SAWeb, RepBath e COTUCA utilizando um limiar de 50% de similaridade. Observa-se para as três bases de estórias em estudo que, apesar do percentual médio de indicações corretas de estórias evoluídas (*precision*) ser de 33,9%

(conforme Tabela 19), o percentual médio de casos reais de evoluções identificadas (*recall*) foi de 75,9% (conforme Tabela 19). Para todas as bases de estórias analisadas, o percentual de estórias evoluídas identificadas, dentre todas as que foram sugeridas como evoluídas, é maior que 70%, o que indica que o método cobriu uma parcela considerável de evoluções que deveriam ter sido apontadas. O valor médio para *recall* obtido pode ser considerado significativo apesar da baixa precisão causada pelos consideráveis números de casos falso positivos para as bases SAWeb e COTUCA. Em geral, o desempenho do método de identificação de estórias evoluídas, considerando a média harmônica entre *recall* e *precision,* foi de 46,1%. A média de acurácia obtida por esse experimento foi 97,4%.

Tabela 18 - Resultado da identificação de estórias que sofrerão evolução para as bases de estórias SAWeb, RepBath e COTUCA com limiar de similaridade em 50%

Base de Estórias	**Número real de evoluções (EVO)**	**Casos de evoluções sugeridos (SUG)**	**Casos Verdadeiro Positivos (VP)**	**Casos Falso Positivos (FP)**	**Casos Verdadeiro Negativos (VN)**	**Casos Falso Negativos (FN)**
SAWeb	15	36	12	24	1446	3
RepBath	8	13	6	7	409	2
COTUCA	11	36	8	28	742	3

Tabela 19 - Valores de *Accuracy*, *Recall* e *Precision* obtidos para as bases SAWeb, RepBath e COTUCA

Base de Estórias	***Recall* (%)** **VP / (VP+FN)**	***Precision* (%)** **VP / (VP+FP)**	***F-measure* (%)** **2 x (*Precision* x *Recall*) / (*Precision* + *Recall*)**	***Accuracy* (%)** **(VP + VN) / (VP+FP+VN+FN)**
SAWeb	80,0%	33,3%	47,0%	98,2%
RepBath	75,0%	46,2%	57,2%	97,9%
COTUCA	72,7%	22,2%	34,0%	96,0%
Médias	75,9%	33,9%	46,1%	97,4%

A Figura 24 representa um gráfico que demonstra o número de indicações verdadeiro e falso positivos no que se refere a indicações de estórias evoluídas, para as bases de estórias SAWeb, RepBath e COTUCA, utilizando o limiar de similaridade no valor de 50%. Nota-se, nesse gráfico, que o número de indicações falso positivos foi maior que o número de indicações verdadeiro positivos com destaque para as bases de estórias SAWeb e COTUCA.

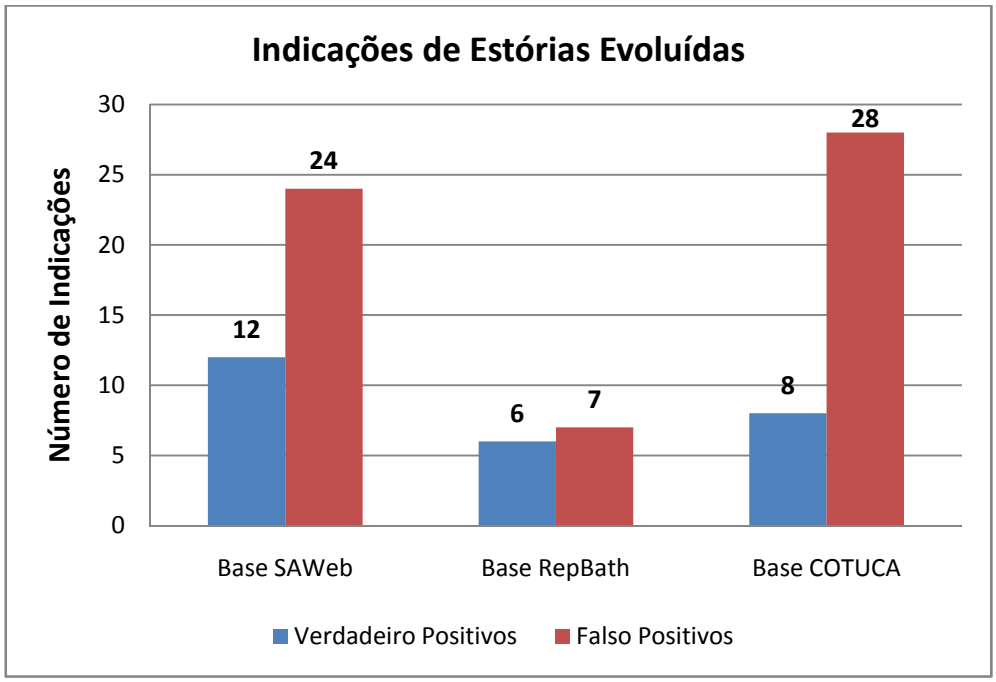


Figura 24 - Gráfico de indicações de estórias evoluídas

A Figura 25 apresenta um gráfico que demonstra os valores de *recall*, *precision, F-measure* e *accuracy* para cada uma das bases de estórias analisadas. Para todas as bases de estórias analisadas, os valores de *recall* foram maiores que o valores para *precision*. Isso demonstra que para tais bases houve uma maior cobertura de identificação de estórias evoluídas dentre todas as estórias de fato evoluídas. Os valores não tão expressivos conseguidos pela medida *F-measure* são consequência dos baixos valores de precisão obtidos. Os valores de acurácia se sobressaem em decorrência do significativo número de casos verdadeiro negativos identificados para as três bases de estórias.

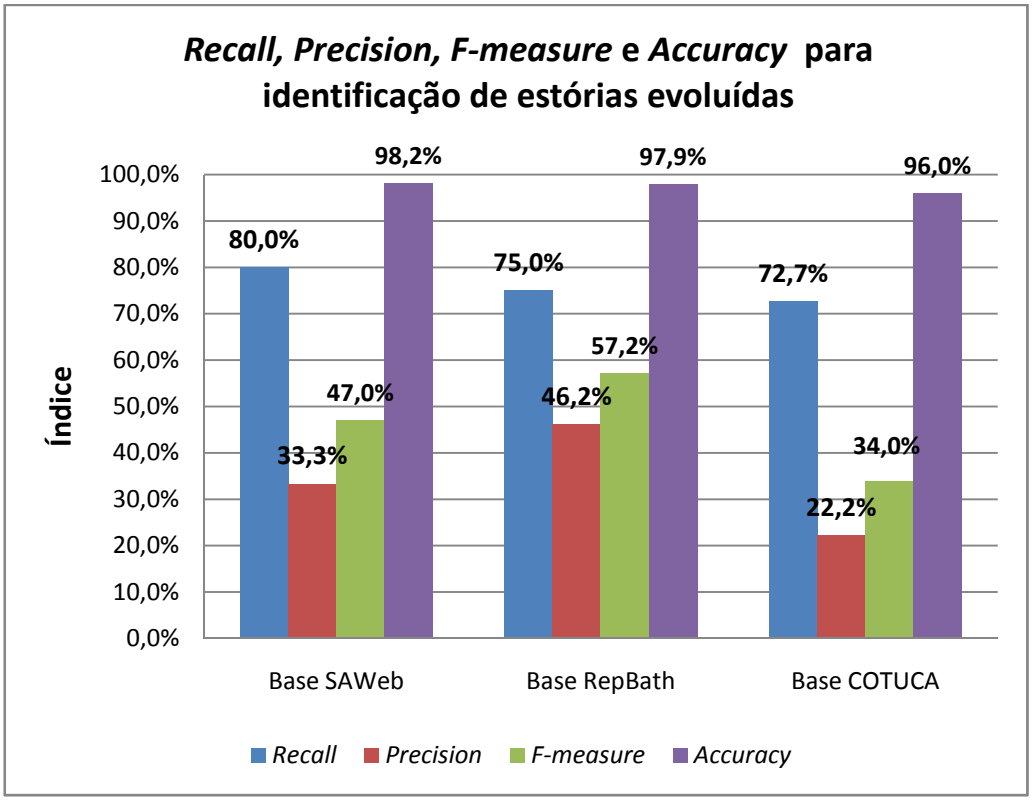


Figura 25 - Médias de *Recall*, *Precision, F-measure* e *Accuracy* para estórias evoluídas

Em um cenário real de uso da metodologia ágil Scrum, é necessário trabalhar com bons níveis de precisão e de cobertura na identificação de estórias evoluídas. Contudo, mais importante que a precisão é a cobertura de indicações corretas de casos de evolução. Altos níveis de cobertura são desejáveis para que os participantes do Scrum tenham maior segurança de que a maioria dos casos de evolução existentes foram descobertos. Dessa forma, os participantes do Scrum terão mais certeza sobre os reais pontos do sistema que serão afetados com a chegada de novas estórias de usuário. Além disso, o conhecimento antecipado do maior número possível de pontos do sistema que sofrerão alteração permitirá que os envolvidos no Scrum tomem decisões mais assertivas sobre a estratégia de desenvolvimento.

7.5 Experimento 4 - Análise de distribuição de estórias em *sprints* por meio de técnica de agrupamento de textos

Nesta seção, será demonstrado o resultado do experimento realizado para a distribuição de estórias de usuário em *sprints* por meio da técnica de agrupamento de textos.

A base de estórias utilizada neste experimento foi a SAWeb. Ela foi analisada por um analista humano, neste caso, um analista de desenvolvimento de sistemas com dois anos de experiência em metodologia ágil Scrum. O analista distribuiu todas as estórias da base SAWeb em nove *sprint*s de quatro semanas. Para cada estória de usuário da base SAWeb, o analista atribuiu uma pontuação referente ao esforço necessário para sua implementação. Para a distribuição de pontos de esforço para cada estória foram adotadas as seguintes pontuações: 1 (minúscula), 2 (muito pequena), 3 (pequena), 5 (média), 8 (grande), 13 (muito grande), 40 (enorme) e 100 (gigante). Essas pontuações foram utilizadas pelo analista de forma a auxiliá-lo na distribuição de estórias em *sprints,* levando em consideração o esforço necessário para a implementação das funcionalidades em um período de quatro semanas.

A Tabela 20 mostra como foram distribuídas manualmente as estórias de usuário da base SAWeb conforme proposta feita pelo analista humano. O *backlog* do produto contendo as estórias de usuário distribuídas em *sprints* pode ser visto no Apêndice D.

Tabela 20 - Distribuição das estórias de usuário em *sprint*s

Sprints	**Total de estórias**	**Total de pontos**
SP1	15	62
SP2	19	59
SP3	13	66
SP4	10	62
SP5	7	58
SP6	15	63
SP7	9	61
SP8	8	57
SP9	3	21
Total	99	509

Para o agrupamento automático das estórias de usuário, foi utilizado o algoritmo de agrupamento particional *k-medoids* (Kaufman & Rousseeuw, 1987). Conforme já mencionado, no algoritmo *k-medoids*, o número de *cluster*s desejados, também conhecido como *k*, é um parâmetro de entrada do algoritmo. O número para o parâmetro *k* informado ao algoritmo *k-medoids* foi o mesmo número de *sprint*s em que as estórias foram manualmente distribuídas pelo analista humano (no caso, nove).

O algoritmo *k-medoids* utilizou a função de similaridade semântica WuP (Wu & Palmer, 1994). Essa função foi utilizada para verificar a similaridade semântica apenas do trecho da estória condizente à funcionalidade, ou seja, os trechos da estória referentes ao tipo de usuário e a razão da funcionalidade não foram considerados. Esses trechos foram desconsiderados durante a análise para evitar qualquer influência nos resultados de similaridade, uma vez que seu conteúdo pode se repetir em outras estórias de usuários.

Para assegurarmos a confiabilidade dos resultados e para evitar a introdução de viés, foram realizadas cem rodadas de execução do algoritmo de agrupamento automático de estórias de usuário. Isso se justifica pela possibilidade do algoritmo produzir resultados de agrupamento discretamente diferentes, já que a seleção das estórias para medoide inicial é feita de forma aleatória. Para cada uma das execuções do algoritmo foram comparadas as distribuições automáticas de estórias com a distribuição manual feita pelo analista humano.

Cada um dos nove *cluster*s gerados em cada uma das rodadas de execução tiveram suas estórias comparadas com a distribuição de estórias manualmente feita pelo analista humano. Nesse processo de comparação, foi computado o número de grupos de estórias que pertenciam a um *sprint* proposto pelo analista humano e a um determinado *cluster* (relacionamento *sprint-cluster*), bem como foi calculado o número de estórias que participam desse relacionamento.

A Tabela 21 apresenta o número médio de relacionamentos *sprint-cluster* distribuídos encontrados por *sprint* a partir da comparação da distribuição manual dos *sprints* com os *clusters* automaticamente gerados por meio da técnica de agrupamento de textos. Os *sprints* e o *total de estórias* referem-se, respectivamente, aos *sprints* propostos pelo analista humano e o número de estórias de usuário para cada *sprint*. O *número médio de relacionamentos sprint-cluster* (M_RSC) representa o número médio de grupos formados por duas ou mais estórias de usuários relacionadas, que foram simultaneamente atribuídas a um *sprint* e também a um *cluster* e é calculado conforme Equação 30, descrita na seção 6.4. Esse

número foi utilizado para verificar a quantidade de porções de estórias relacionadas que foram detectadas para cada *sprint*. O *número médio de estórias em clusters independentes* (M_ECI) indica a média aritmética do número de estórias de um *sprint* que foram separadas e atribuídas individualmente a *clusters* distintos e, é calculado conforme Equação 33, descrita na seção 6.4. O *número médio de estórias por relacionamento sprint-cluster* indica a média aritmética do número de estórias que foi encontrado em cada relacionamento *sprint-cluster* e, é calculado conforme Equação 31, descrita na seção 6.4. Esse número é um indicativo de tamanho do relacionamento *sprint-cluster.* O *percentual de estórias relacionadas* refere-se a parcela que o número médio de estórias por relacionamento *sprint-cluster* representa em seu respectivo *sprint*.

Tabela 21 - Relacionamentos *sprint-cluster* encontrados

Sprints **(A)**	**Total de estórias** **(B)**	**Número médio de relacionamentos *sprint-cluster*** **(M_RSC)**	**Número médio de estórias por relacionamento *sprint-cluster*** **(M_EST)**	**Número médio de estórias em *clusters* independentes** **(M_ECI)**	**Percentual de estórias relacionadas por *sprint*** **(C) = M_EST / B**
SP1	15	4	3	3	20,0%
SP2	19	5	3	4	15,8%
SP3	13	3	2	7	15,4%
SP4	10	2	2	6	20,0%
SP5	7	2	2	3	28,6%
SP6	15	3	3	6	20,0%
SP7	9	2	2	5	22,2%
SP8	8	1	2	6	25,0%
SP9	3	0	0	3	0,0%
Médias		2,4	2,1	4,8	18,6%

Nota-se na Tabela 21 que, os relacionamentos *sprint-clusters* detectados possuem de 2 a 3 estórias de usuário relacionadas, ou seja, existem 2 a 3 estórias de um *sprint* que possuem algum relacionamento em seu trecho de funcionalidade detectado por similaridade semântica. Dependendo do número de estórias que um *sprint* possua, esse relacionamento entre as estórias pode representar um percentual significativo do *sprint*, como é o caso do *sprint* 5, em que o percentual do *sprint* que o relacionamento representa, chega a 28,6%. Percebe-se, no caso do *sprint* 9, que não foram encontrados casos de relacionamentos *sprint-cluster*. Isso pode ser um indicativo que o analista humano atribuiu a esse *sprint* estórias de usuário com funcionalidades isoladas ou independentes das demais.

A Figura 26 apresenta um gráfico em que se é possível visualizar o número de estórias relacionadas por *sprint*.

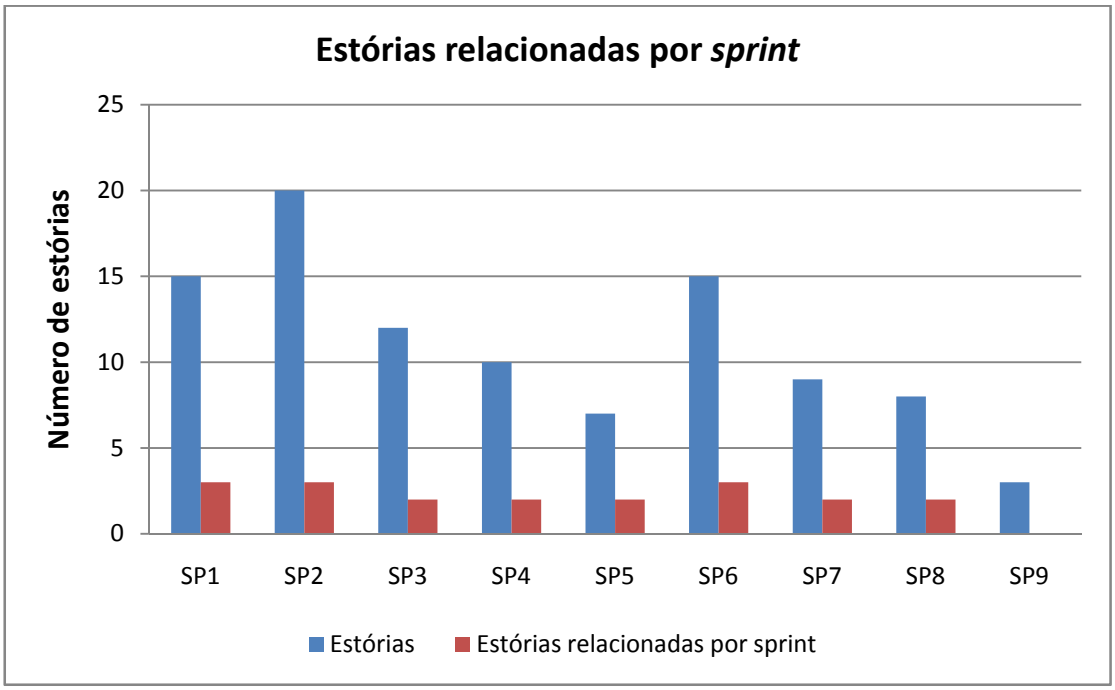


Figura 26 - Número de estórias relacionadas por *sprint*

O resultado do experimento demonstra que o agrupamento de estórias de usuário por similaridade semântica pode revelar como estão relacionadas as estórias dentro de um *sprint*. Nota-se, na Figura 26, que, em geral, os *sprints* propostos pelo analista humano apresentam estórias de usuário pouco relacionadas, o que pode ser constatado pelo percentual médio de estórias relacionadas nos *sprints*; nesse caso, 18,6%, conforme Tabela 21. Em um cenário real de uso da metodologia ágil Scrum, a constatação de *sprints* com estórias pouco relacionadas pode indicar aos envolvidos no planejamento do *sprint* que um módulo do sistema pode não ser totalmente entregue no *sprint* atual, visto que uma ou mais funcionalidades foram atribuídas à *sprints* futuros.

O *sprint* 2, por exemplo, com seus 5 relacionamentos *sprint-cluster* distribuídos, apresenta-se como o *sprint* mais fracionado (em termos de estórias relacionadas), em que cada fração de estórias relacionadas representa apenas 15,8% do total de estórias do *sprint*. No Scrum, um *sprint* tão fracionado pode ser um indicativo de que decidiu-se por atender

diferentes módulos de um sistema, o que pode (ou não) ser uma boa estratégia dependendo do valor de negócio que se deseja entregar com o *sprint*.

A Figura 27 apresenta um trecho do relatório de análise de *backlog* do produto gerado pela ferramenta de apoio à decisão desenvolvida neste trabalho. Neste trecho é apresentada uma análise mais detalhada do *sprint* com o maior número de relacionamentos *sprint-cluster*, no caso, o *sprint* 2.

```
SPRINT 2 (19 stories)
Stories distributed in 9 cluster(s)
AVG similarity between stories 0.14148514
Sprint-Cluster Relationship (SCR):
        - SCR 1: 4 stories (21.052631%) are related (stories 35, 66, 76, 85 are in cluster 2)
        - SCR 2: 3 stories (15.789473%) are related (stories 7, 34, 49 are in cluster 5)
        - SCR 3: 2 stories (10.526316%) are related (stories 4, 46 are in cluster 6)
        - SCR 4: 2 stories (10.526316%) are related (stories 15, 58 are in cluster 8)
        - SCR 5: 4 stories (21.052631%) are related (stories 52, 57, 60, 68 are in cluster 0)
        - story 30 (5.263158%)  is in cluster 3
        - story 47 (5.263158%)  is in cluster 1
        - story 67 (5.263158%)  is in cluster 7
        - story 74 (5.263158%)  is in cluster 4
```

Figura 27 - Relatório parcial sobre análise de *backlog* (destaque para o *sprint* 2)

Nota-se, na Figura 27, que as estórias de usuário do *sprint* 2 apresentam uma média de similaridade de aproximadamente 14,1%, o que é um indicativo de que o *sprint* como um todo possui estórias com baixo nível de relacionamento. Na Figura 27, é apresentada a detecção de 5 relacionamentos *sprint-cluster*. Esses relacionamentos representam fragmentos do *sprint*; nesse caso, estórias de usuário cuja correlação foi confirmada por um *cluster.* Nota-se que, quatro estórias de usuário (estórias 30, 47, 67 e 74) não possuem relacionamento com as demais estórias do *sprint*, o que foi confirmado pelo resultado do agrupamento que sugeriu que elas ficassem em *clusters* independentes.

A Figura 28 apresenta um gráfico em que se vê todos os relacionamentos *sprint-cluster* por *sprint* e como cada um deles está fracionado. Nota-se, que os *sprints* podem ser compostos por parcelas de estórias que estão relacionadas por funcionalidade. Os *sprints* 1 e 2, por exemplo, são os que apresentaram o maior número de parcelas de estórias relacionadas, o que sugere que estes *sprints* iniciais priorizaram um número maior de módulos do sistema, possivelmente para se entregar incrementos de produto com mais valor de negócio logo nos primeiros ciclos do projeto. Percebe-se também que, os *sprints* 8 e 9, são os que apresentaram o menor número de relacionamentos *sprint-cluster*, no caso, 1 e 0, respectivamente. Para estes

sprints, o número reduzido de relacionamentos *sprint-cluster* é consequência de serem os *sprints* finais do projeto, em que nas entregas finais é comum que os incrementos de produto possuam funcionalidades isoladas e pouco relacionadas, já que se tratam das funcionalidades restantes e de mais baixa prioridade.

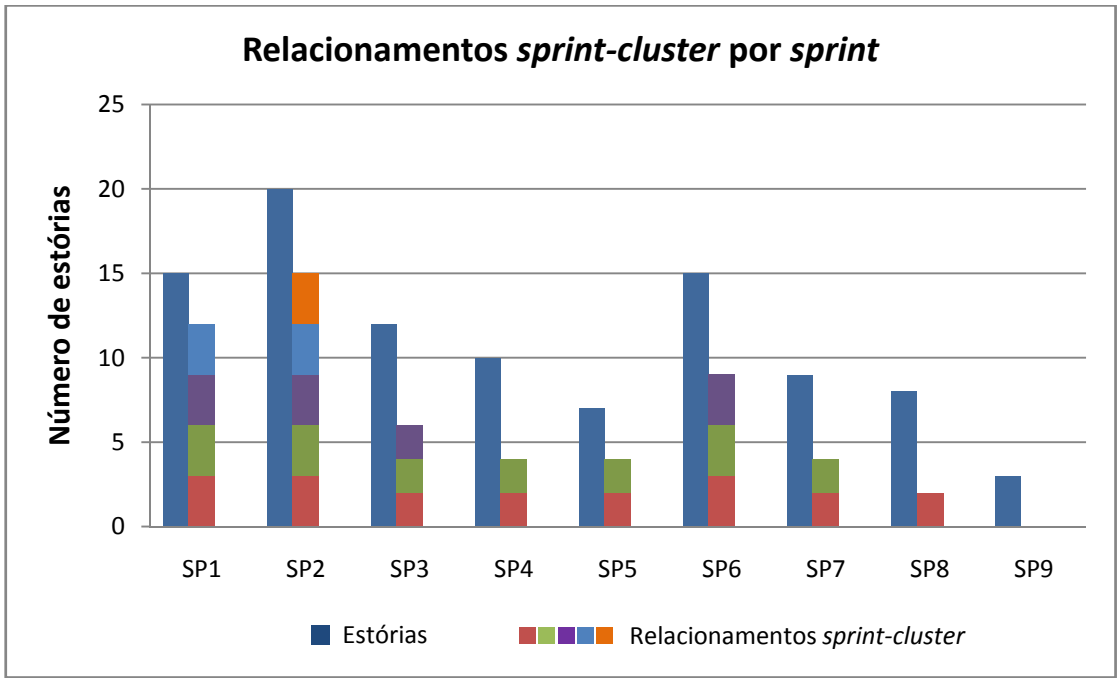


Figura 28 - Relacionamento *sprint-cluster* por *sprint*

Os relacionamentos encontrados a partir da comparação de um *sprint* proposto por um analista humano e o agrupamento automático de estórias por similaridade podem justificar ou mesmo orientar um analista de requisitos a selecionar estórias de usuário com funcionalidades mais relacionadas para a composição de um *sprint*. As informações inferidas a partir destes relacionamentos são úteis para o dono do produto e para a equipe de desenvolvimento Scrum, pois durante o planejamento do *sprint* eles conseguirão visualizar como estão relacionadas as funcionalidades em um *sprint*, auxiliando-os na formação de *sprints* com maior valor entregável. O desenvolvimento de *sprint*s com funcionalidades mais interligadas permitirá o aumento da velocidade de implementação, além de acelerar o aprendizado do sistema pelo time de desenvolvimento.

Este capítulo apresentou as bases de dados, as funções de similaridade e o algoritmo de agrupamento utilizados neste trabalho. Este capítulo também apresentou os

resultados obtidos com os experimentos realizados. Dentre as funções de similaridade analisadas, a função de similaridade WuP foi a que apresentou os melhores índices de precisão e sensibilidade quando comparada com outras funções estudadas. No que se refere a indicação de estórias duplicadas, o uso da função de similaridade semântica permitiu que fossem descobertos um considerável montante de casos reais de estórias duplicadas. Além disso, o método proposto para a identificação de estórias duplicadas obteve altos índices de acurácia. Quanto a identificação de estórias que evoluirão, o experimento realizado indicou um alto nível de cobertura e acurácia, apesar dos baixos índices de precisão alcançados. Para a identificação de duplicidades e evolução em estórias, as medidas *recall*, *precision*, *F-measure* e *accuracy* permitiram que os resultados obtidos pudessem ser bem interpretados. O experimento para análise de distribuição de estórias em *sprints* mostrou que o agrupamento de estórias por similaridade pode confirmar a escolha feita por um analista humano sobre as estórias que comporão um *sprint* do Scrum.

8 Ferramenta

A ferramenta desenvolvida neste trabalho utiliza os métodos apresentados até então e se destina a apoiar as tarefas de análise de estórias de usuário para identificação de duplicidades, evolução de requisitos e composição de *sprints*. A ferramenta gera informações sobre as estórias de usuário, para auxiliar *Scrum Masters,* desenvolvedores e demais envolvidos a identificar estórias de usuários com funcionalidades duplicadas, identificar estórias que sofrerão alguma evolução e, por fim, auxiliar na confirmação da escolha de estórias de usuário para determinados *sprints* de desenvolvimento.

Nas próximas subseções, será apresentada a ferramenta e sua utilização no Scrum, sua arquitetura e organização, e, por fim, suas funcionalidades.

8.1 A ferramenta e sua utilização no Scrum

A ferramenta proposta neste trabalho, chamada CLASS (acrônimo de *CLustering and Analysis of Scrum Stories*), pode ser utilizada durante a reunião de refinamento do *backlog* do produto (do Inglês, *backlog grooming* - livre tradução) para auxiliar a identificação de duplicidades e evoluções em estórias de usuário e ser utilizada na fase do planejamento do *sprint* para confirmar a composição de *sprints* ou mesmo propor alterações para otimizá-los.

A ferramenta auxilia os participantes do *backlog grooming* a identificarem estórias duplicadas no momento da chegada de novas estórias, sugerindo possíveis casos de duplicidades de estórias com base no nível de similaridade encontrado entre as novas estórias de usuário e as estórias existentes no *backlog do produto*. A ferramenta também apoia os participantes do *backlog grooming* a identificarem as estórias do *backlog* do produto (ou estórias já implementadas) que serão alteradas por novas estórias de usuários, isto é, estórias que sofrerão transformação ou evolução. A abordagem utilizada para identificação de evolução de estórias também se baseia na análise de similaridade textual entre as estórias de usuário.

Na fase de planejamento do *sprint*, a ferramenta realiza o agrupamento de estórias de usuário por similaridade para apoiar os participantes a verificarem o relacionamento das

estórias de usuário dentro de um grupo. Para essa situação, o conhecimento sobre o relacionamento entre as estórias de usuário é útil para otimizar o desenvolvimento do produto e apoiar a equipe de desenvolvimento na inclusão de uma estória em um dado *sprint.* O agrupamento de estórias auxilia os participantes do planejamento do *sprint* a confirmarem se um *sprint* está bem formado e se é necessário realocar estórias entre *sprints.*

A Figura 29 mostra o *framework* Scrum e os eventos em que as funcionalidades da ferramenta proposta são utilizadas.

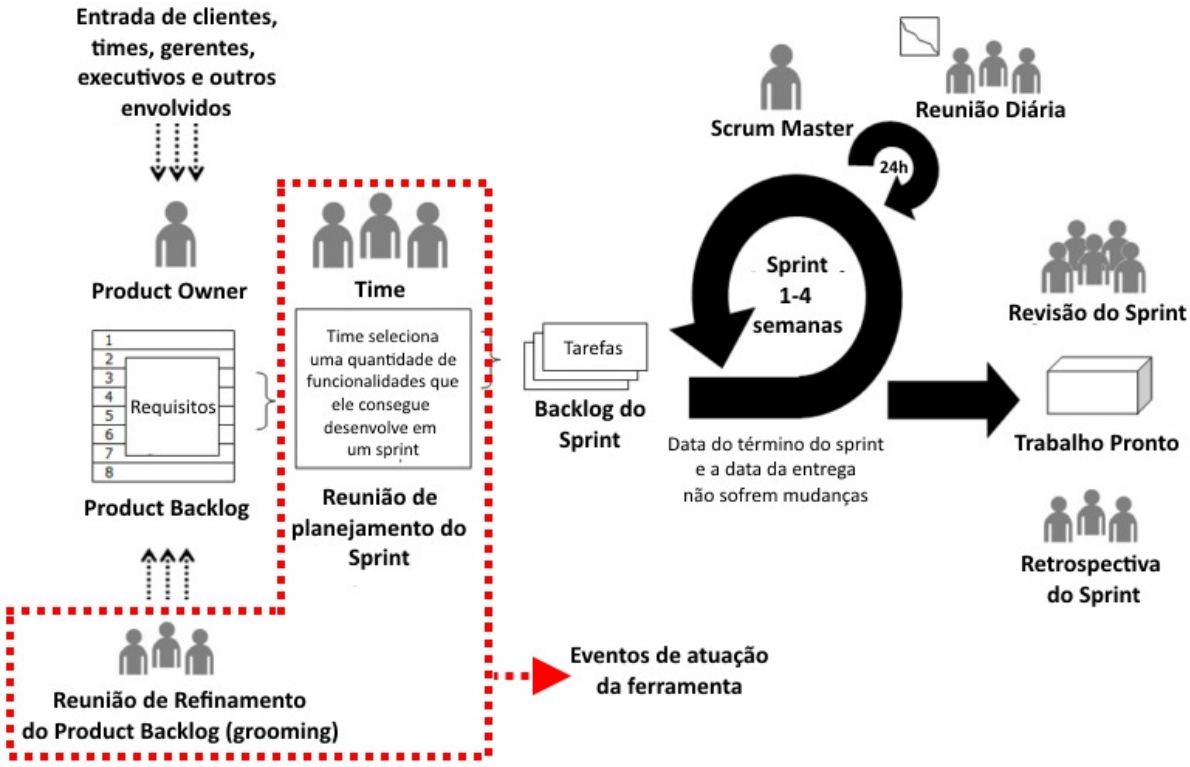


Figura 29 - *Framework* Scrum (traduzido e adaptado de Sutherland e Schwaber, 2007)

8.2 Organização e arquitetura da ferramenta

A ferramenta foi desenvolvida em linguagem de programação Java e faz uso de bibliotecas auxiliares como, por exemplo, JFreeChart (Viklund, 2005), MDSJ – *Multidimensional Scaling for Java* (Pich., 2009) e SEMILAR (Rus et al., 2013). A biblioteca auxiliar JFreeChart é uma biblioteca de código aberto utilizada para a criação de gráficos personalizados utilizando a linguagem Java. A biblioteca auxiliar MDSJ é uma

implementação livre e não gráfica de algoritmos básicos de *Multidimensional Scaling,* podendo ser utilizada tanto como um aplicativo independente ou como um módulo em aplicações Java para análise e visualização de dados. Por fim, biblioteca auxiliar SEMILAR disponibiliza um conjunto de funções de similaridade, além de métodos que permitem a extração de características sintáticas e léxicas de textos.

A ferramenta implementa o algoritmo de agrupamento *k-medoids* (Kaufman & Rousseeuw, 1987) para a realização do agrupamento das estórias de usuário. A função de similaridade semântica WuP (Wu & Palmer, 1994) foi empregada para a identificação de semelhanças entre as estórias de usuário. As técnicas de Processamento de Linguagem Natural (PLN), remoção de *stopwords*, tokenização, lematização e estemização também foram empregadas na ferramenta. A Figura 30 apresenta a arquitetura da ferramenta desenvolvida neste trabalho.

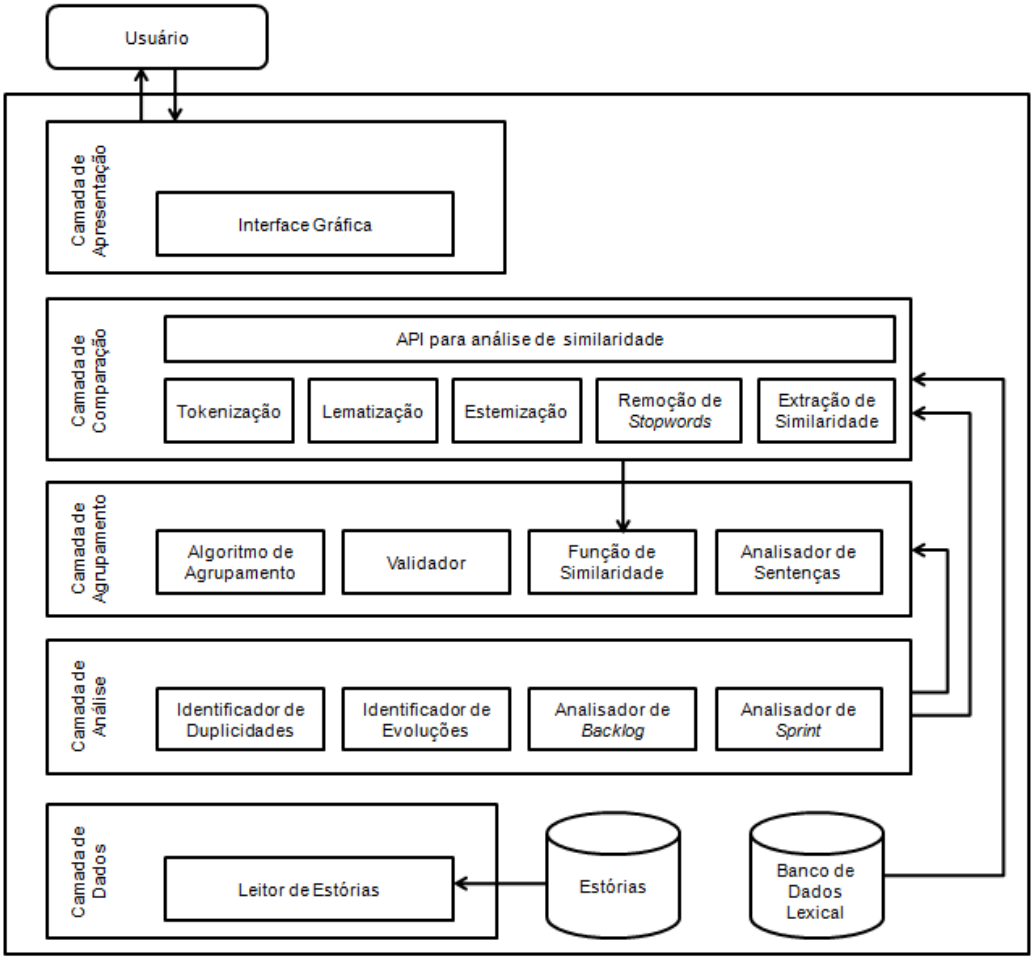


Figura 30 - Arquitetura da ferramenta

A ferramenta apresenta uma *camada de apresentação,* que é responsável pela interação do usuário com a ferramenta; uma *camada de comparação,* que realiza o processamento e extração da similaridade das estórias; uma *camada de agrupamento,* que cabe a extração de grupos de estórias semelhantes; uma *camada de análise* que verifica as estórias de usuário para identificar casos de duplicidades e evolução além de analisar o *backlog* do produto e *sprints* para sugerir alterações para otimização; e uma *camada de dados,* que é responsável pela leitura das estórias de usuário.

A camada de apresentação representa as interfaces gráficas que permitem ao usuário interagir com a ferramenta. Essa camada é composta por interfaces que permitem realizar as operações na ferramenta, como, exibir os resultados, agrupar estórias, sugerir casos de duplicidades e evoluções em estórias de usuários, e solicitar análise de *backlog* e *sprint.*

A camada de comparação é responsável por comparar as estórias de usuário e verificar a similaridade existente entre elas. Essa camada utiliza o *toolkit* SEMILAR (Rus et al., 2013) e a base de dados lexical WordNet (Miller, 1995) para extração de informações que são utilizadas pela função de similaridade semântica WuP (Wu & Palmer, 1994).

A camada de agrupamento realiza o agrupamento das estórias de usuário por meio do algoritmo *k-medoids* e utiliza um módulo de validação baseado no coeficiente de Silhueta para realizar a análise do agrupamento gerado e indicar a qualidade do agrupamento gerado. Para detectar o papel sintático das palavras que compõem as estórias de usuário, essa camada conta com o analisador (*parser*) de linguagem natural da Universidade de Stanford (De Marneffe et al., 2006).

A camada de análise permite analisar o *backlog* do produto, para identificar possíveis casos de duplicidade e evolução em estórias de usuário. Ela também permite analisar o *backlog* do produto e sugerir alterações em *sprints* com base em um agrupamento realizado e nas semelhanças encontradas entre as estórias de usuário.

Por fim, a camada de dados é utilizada pela ferramenta para fazer a leitura das estórias de usuário. Essa camada é responsável por ler um conjunto de estórias de usuário e disponibilizá-las para as demais camadas da ferramenta. Ela também é responsável por permitir que sejam realizadas operações no conjunto de estórias de usuário, como, por exemplo, inclusão, exclusão e edição de estórias de usuário.

A Figura 31 apresenta o diagrama de caso de uso da ferramenta desenvolvida neste trabalho.

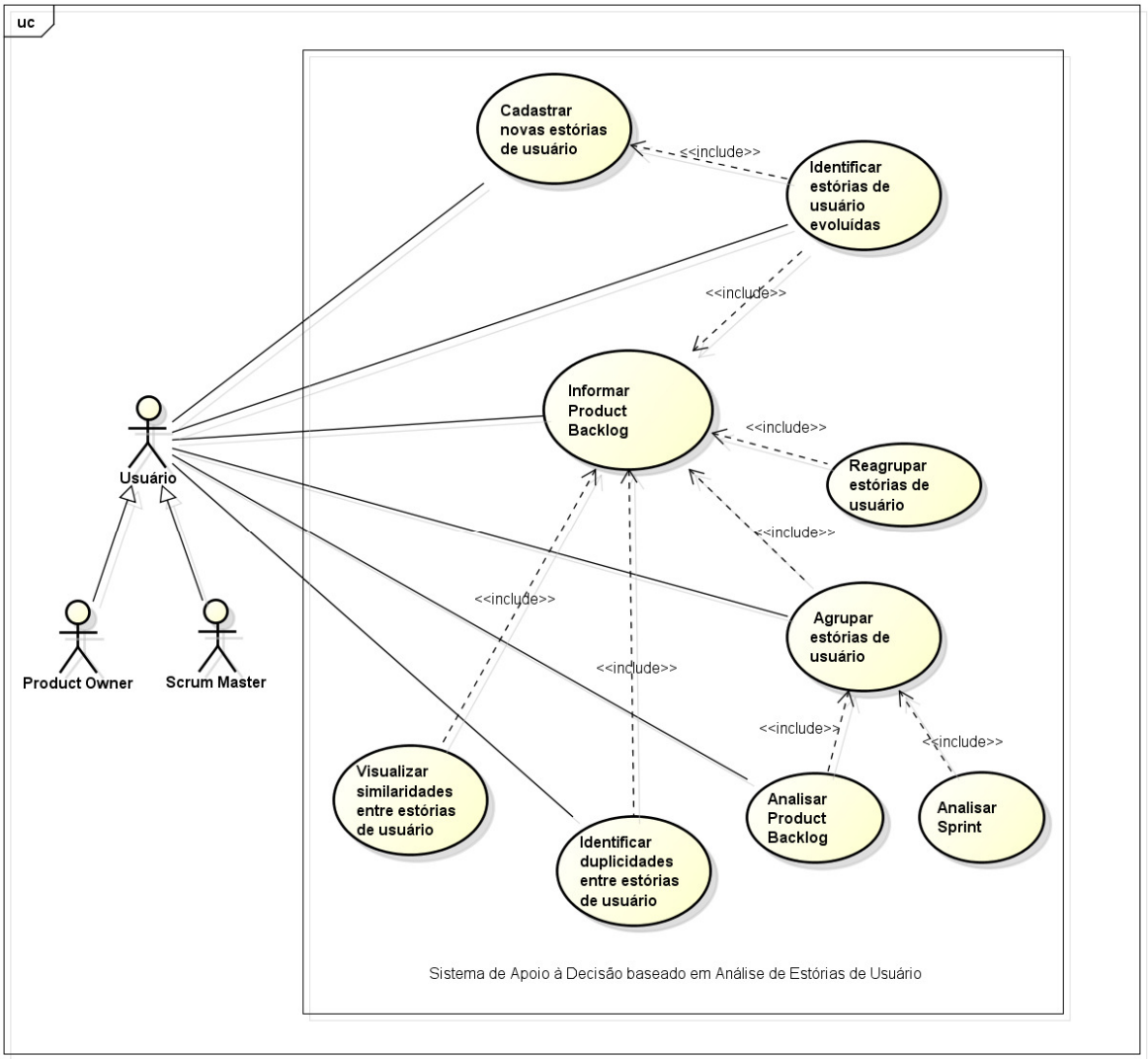


Figura 31 - Diagrama de caso de uso da ferramenta

O caso de uso *Informar Product Backlog* refere-se à funcionalidade que permite ao usuário carregar um *backlog* do produto para ser analisado pela ferramenta. O caso de uso *Cadastrar novas estórias de usuário* refere-se à funcionalidade que possibilita ao usuário informar as novas estórias de usuário que surgirem durante o desenvolvimento de um sistema. Após a entrada de novas estórias de usuário, elas são comparadas com as estórias que permanecem no *backlog* do produto e com as estórias já removidas do *backlog* do produto (estórias já implementadas). A comparação das novas estórias com as estórias do *backlog* do produto permite a sugestão de estórias duplicadas, ao passo que a comparação das novas

estórias com as estórias já implementadas permite a sugestão de estórias que sofrerão evolução.

O caso de uso *Agrupar estórias de usuário* agrupa as estórias de usuário em diferentes conjuntos, com o uso de um algoritmo de agrupamento em conjunto com uma função de similaridade. A Figura 32 apresenta a tela inicial da ferramenta após o agrupamento de estórias de usuário contidas em um *backlog* do produto.

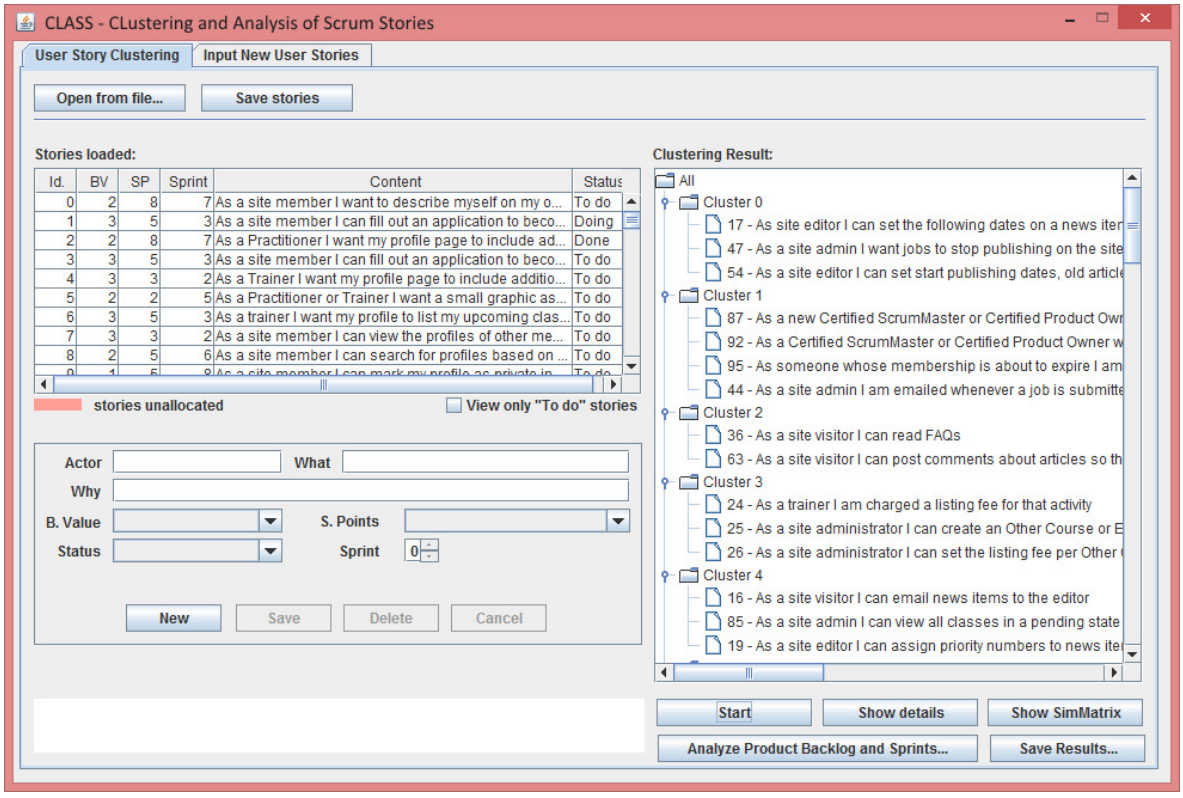


Figura 32 - Tela inicial da ferramenta após o agrupamento de estórias de usuário

Após o agrupamento das estórias de usuário, a ferramenta disponibiliza informações como similaridade entre estórias de usuário e contexto dos grupos formados. Já o caso de uso *Reagrupar estórias de usuário* permite que as estórias existentes no *backlog* do produto sejam reagrupadas, a fim de identificar novos relacionamentos a partir da chegada de novas estórias de usuário.

O caso de uso *Visualizar similaridade entre estórias de usuário* refere-se à funcionalidade da ferramenta que apresenta ao usuário uma matriz de similaridade formada a partir das estórias de usuário analisadas. A matriz de similaridade gerada pela ferramenta

apresenta os valores de similaridade extraídos a partir de cada combinação formada de pares de estórias de usuário.

O caso de uso *Identificar duplicidades entre estórias de usuário* refere-se à funcionalidade que sugere possíveis casos de estórias duplicadas que possam existir em um *backlog* do produto. A ferramenta analisa os níveis de similaridade existentes entre as novas estórias de usuário e as estórias existentes no *backlog* do produto. Caso os níveis de similaridade atinjam um limiar informado pelo usuário, a ferramenta sugerirá potenciais casos de duplicidades entre estórias que, posteriormente, serão confirmados pelo dono do produto e *Scrum Master.* A Figura 33 apresenta exemplos de casos de duplicidade sugeridos pela ferramenta a partir do limiar de similaridade de 70%, em que *New* representa uma nova estória que será adicionada ao *backlog* do produto e PB (*Product Backlog*) uma estória já existente no *backlog* do produto.

```
CANDIDATES TO DUPLICATION (Similarity >= 70%)

  New: 0 -  As a member I can round out an application to end up a Practitioner
  PB: 1 - As a site member I can fill out an application to become a Practitioner
  Sim(0,1) = 0.8888889

  New: 1 - As a visitor I can round out an application to end up a Trainer
  PB: 3 - As a site member I can fill out an application to become a Trainer
  Sim(1,3) = 0.8888889

  New: 3 -  As a administrator I can alter any site part profile
  PB: 13 - As a site administrator I can edit any site member profile
  Sim(3,13) = 0.95238096

  New: 4 -  As a member I can get to old news that is no more on the landing page
  PB: 14 - As a site visitor I can read current news on the home page
  Sim(4,14) = 0.7058824

  New: 4 -  As a visitor I can get to old news that is no more on the landing page
  PB: 15 - As a site visitor I can access old news that is no longer on the home page
  Sim(4,15) = 0.8067227

  New: 19 -  As a administrator I can do a full-content hunt of article body, title, and writer name
  PB: 61 - As a site visitor I can do a full-text search of article body, title, and author name
  Sim(19,61) = 0.85714287
```

Figura 33 - Casos de duplicidade sugeridos pela ferramenta

O caso de uso *Identificar estórias de usuário evoluídas* trata da funcionalidade que sugere possíveis casos de estórias de usuário que sofrerão evolução ou alteração após a entrada de novas estórias de usuário. Durante o processo de comparação, a ferramenta analisa os níveis de similaridade existentes entre as novas estórias e as estórias já implementadas. Caso os níveis de similaridade atinjam o limiar informado pelo usuário, a ferramenta sugerirá

potenciais casos de estórias que evoluirão e que, posteriormente, serão confirmados pelo dono do produto e *Scrum Master.* A Figura 34 apresenta um exemplo de casos de evolução sugeridos pela ferramenta a partir do limiar de similaridade de 50%, em que *New* representa uma nova estória que será adicionada ao *backlog* do produto e *Evolution* a estória candidata a passar por evolução.

CANDIDATES TO EVOLUTION (Similarity >= 50%)

New: 5 - As a site administrator I can edit only Practitioner profile
Evolution: 1 - As a site member I can fill out an application to become a Practitioner
Sim(5,1) = 0.5555556

New: 0 - As a Practitioner I want to include in my profile details about the courses that I attended
Evolution: 2 - As a Practitioner I want my profile page to include additional details about me
Sim(0,2) = 0.6

New: 0 - As a Practitioner I want to include in my profile details about the courses that I attended
Evolution: 4 - As a Trainer I want my profile page to include additional details about me
Sim(0,4) = 0.6

New: 1 - As a site member I want to view the profiles of trainers the same way I see the profiles of members
Evolution: 7 - As a site member I can view the profiles of other members
Sim(1,7) = 0.6666667

New: 5 - As a site administrator I can edit only Practitioner profile
Evolution: 7 - As a site member I can view the profiles of other members
Sim(5,7) = 0.6181818

Figura 34 - Casos de evolução sugeridos pela ferramenta

O caso de uso *Analisar Product Backlog* realiza uma comparação entre os *sprints* inicialmente propostos pelo usuário e *clusters* gerados pelo processo de agrupamento. Esse tipo de análise permite ao dono do produto, *Scrum Master* e ao time de desenvolvimento visualizarem, durante a reunião de planejamento de *sprint,* como estão relacionadas as estórias de usuário dentro do *sprint* inicialmente proposto. Esse caso de uso permite aos envolvidos visualizarem o percentual das estórias de um *cluster* que pertence a um determinado *sprint*.

O caso de uso *Analisar Sprint* analisa o *sprint* informado pelo usuário e sugere possíveis alterações para o *sprint* com base no resultado de agrupamento das estórias. Esta funcionalidade verifica a similaridade média entre as estórias e o esforço total para a implementação do *sprint* e, quando possível, sugere a remoção e/ou inserção de novas estórias, com intenção de aumentar a similaridade interna do *sprint*. Nesse processo, também

é verificado o valor de negócio (do Inglês, *Business Value*) do *sprint,* a fim de sugerir alterações que aumentem o valor de negócio total e consequentemente o valor agregado do incremento do produto.

O Quadro 11 apresenta um algoritmo preliminar utilizado para sugerir alterações no *sprint* a partir do agrupamento automático de estórias de usuário. O analista humano informa ao algoritmo qual *sprint* se deseja analisar e o número máximo de pontos de estória que o time se compromete a implementar. Em seguida, o algoritmo identifica o *cluster* mais parecido ($cluster_{base}$) com o *sprint* a ser analisado e, a partir dele, acrescenta ou remove estórias, para propor um *sprint* mais coeso e com maior valor de negócio total. Para isso, em caso de sugestão de um novo *sprint*, o algoritmo selecionará em ordem a estória de usuário com maior valor de negócio (função *estoriaComMaiorValorDeNegocio*) e a adicionará ao novo *sprint* que será sugerido. O processo de adição de estórias ao novo *sprint* ocorrerá enquanto for possível adicionar novas estórias respeitando o limite máximo de pontos de estória a ser implementado, no caso, TSP_{max}.

Quadro 11 - Algoritmo para sugestão de alterações em *sprint*

entrada: - *sprint* a ser analisado *S*
- total de pontos de estória máximo para ser implementado TSP_{max}

saída: - sugestão *sprint S'*

$cluster_{base}$ = *clusterQueMaisContemEstóriasDe*(*S*)
se (*totalDeStoryPoints*($cluster_{base}$) = *totalDeStoryPoints*(*S*)) **então**
    S' = estoriasDe($cluster_{base}$)
senão
    se (*totalDeStoryPoints*($cluster_{base}$) < *totalDeStoryPoints*(*S*)) **então**
        S' = *estoriasDe*($cluster_{base}$)
        conjunto = *S* - $cluster_{base}$
        enquanto (*totalDeStoryPoints*(*S'*) < TSP_{max} **E** *conjunto* ≠ ∅) **faça**
            estoria = *estoriaComMaiorValorDeNegocio*(*conjunto*)
            conjunto = *conjunto* - *estoria*
            S' = *estoriasDe*(*S'*) + *estoria*
        fimenquanto
    senão
        S' = $cluster_{base} \cap S$
        conjunto = $cluster_{base}$ - *S*
        enquanto (*totalDeStoryPoints*(*S'*) < TSP_{max} **E** *conjunto* ≠ ∅) **faça**
            estoria = *estoriaComMaiorValorDeNegocio*(*conjunto*)
            conjunto = *conjunto* - *estoria*
            S' = *estoriasDe*(*S'*) + *estoria*
        fimenquanto
    fimse
fimse

A Figura 35 apresenta um exemplo de sugestões de alterações para um *sprint*. Nele, as modificações sugeridas para o *sprint* 7 foram a remoção das estórias de usuário 82 e 84 e a inclusão das estórias de usuário 24 e 26. Percebe-se que o *sprint* 7 originalmente possuía o valor de negócio total 17, mas que, com as alterações propostas, passou a possuir o valor de negócio total 21. Além disso, a similaridade interna do *sprint* passou de 17,65% para 20,48%. Com as alterações propostas, o *sprint* 7 passou a possuir maior valor de negócio total e a tratar de estórias de usuário mais relacionadas.

ANALYSIS OF SPRINT

Sprint to be analyzed: *Sprint* 7
Number of stories to be maintained: 3
Maximum number of points for this *sprint*: 61

CURRENT SPRINT
Sprint Id.: 7
  User story: 0 - 8 pts - As a site member I want to describe myself on my own page in a semi-structured way
  User story: 2 - 8 pts - As a Practitioner I want my profile page to include additional details about me
  User story: 10 - 3 pts - As a site member I can mark my email address as private even if the rest of my profile is not
  User story: 25 - 8 pts - As a site administrator I can create an Other Course or Event that is not charged a listing ...
  User story: 29 - 8 pts - As a trainer I can copy one of my courses or events so that I can create a new one
  User story: 42 - 8 pts - As someone who wants to hire I can post a help wanted ad
  User story: 80 - 8 pts - As a site visitor I want there to be a section of the website that teaches me the basics of what Scrum is
  User story: 82 - 5 pts - As a site visitor I can view lists on the site of all Certified ScrumMasters, Practitioners, Trainers, and Certified *Product* Owners
  User story: 84 - 5 pts - As a trainer who has finished teaching a Certification class I can load an Excel file into the site
Total Points: 61
Total Value: 17
AVG Sim Intra *Sprint*: 0.17655334

SUGGESTED SPRINT
Sprint Id.: 1
  User story: 24 - 5 pts - As a trainer I am charged a listing fee for that activity
  User story: 25 - 8 pts - As a site administrator I can create an Other Course or Event that is not charged a listing fee ...
  User story: 26 - 5 pts - As a site administrator I can set the listing fee per Other Course or Event
  User story: 0 - 8 pts - As a site member I want to describe myself on my own page in a semi-structured way
  User story: 2 - 8 pts - As a Practitioner I want my profile page to include additional details about me
  User story: 29 - 8 pts - As a trainer I can copy one of my courses or events so that I can create a new one
  User story: 42 - 8 pts - As someone who wants to hire I can post a help wanted ad
  User story: 80 - 8 pts - As a site visitor I want there to be a section of the website that teaches me the basics of what Scrum is
  User story: 10 - 3 pts - As a site member I can mark my email address as private even if the rest of my profile is not
Total Points: 61
Total Value: 21
AVG Sim Intra *Sprint*: 0.20482954

MODIFICATIONS:

Stories (Out):
  User story: 82 - 5 pts - As a site visitor I can view lists on the site of all Certified ScrumMasters, Practitioners, Trainers, and Certified *Product* Owners
  User story: 84 - 5 pts - As a trainer who has finished teaching a Certification class I can load an Excel file into the site
Total points: 10
Total value: 4

Stories (In):
  User story: 24 - 5 pts - As a trainer I am charged a listing fee for that activity
  User story: 26 - 5 pts - As a site administrator I can set the listing fee per Other Course or Event
Total points: 10
Total value: 8

Figura 35 - Sugestões de alterações em *sprint* propostas pela ferramenta

O algoritmo apresentado no Quadro 11 trata-se de uma proposta preliminar que será validada em trabalhos futuros por experimentos controlados que indicarão seu nível de eficácia.

Este capítulo apresentou a ferramenta CLASS desenvolvida neste trabalho. Esta ferramenta implementa um algoritmo de agrupamento e função de similaridade, além dos algoritmos de análise de estórias para a identificação de casos de duplicidade e evolução em estórias de usuário. A ferramenta também implementa um algoritmo preliminar para a sugestão de alterações em *sprint* a partir do resultado do agrupamento de estórias de usuário. A ferramenta desenvolvida neste trabalho permite os envolvidos no Scrum obterem informações úteis para a tomada de decisão em atividades realizadas na reunião de refinamento do *backlog* do produto e na reunião de planejamento de *sprints*.

9 Conclusão

Apesar do dinamismo da metodologia Scrum e do forte estímulo à interação entre seus usuários, o Scrum não está livre de problemas comuns do gerenciamento de requisitos como, por exemplo, identificação de requisitos duplicados e identificação e acompanhamento de requisitos modificados. Este trabalho apresentou uma abordagem que utiliza a técnica de agrupamento de textos, técnicas de processamento de linguagem natural e análise semântica para analisar estórias de usuário, a fim de sugerir casos de duplicidades e evolução entre estórias e auxiliar os participantes do Scrum a desenvolverem *sprint*s com funcionalidades semelhantes.

Os experimentos realizados neste trabalho demonstraram que a verificação de similaridade semântica entre estórias de usuário é uma abordagem viável para a identificação de duplicidade e evolução em requisitos desse tipo. Apesar da existência de falsos positivos, para a sugestão de duplicidades e evolução de estórias, a abordagem proposta por este trabalho conseguiu identificar boa parte dos casos de duplicidade e evolução gerados para os experimentos e, portanto, é uma abordagem efetiva e promissora para tais tarefas. O alto percentual de acurácia para a sugestão de duplicidades e evolução deu-se em consequência do elevado número de casos verdadeiro negativos apontado pelos experimentos. Isso é um indicativo de que há uma relação entre similaridade textual de requisitos e a presença de relacionamentos dos tipos duplicação e evolução.

Acredita-se que a existência de falsos positivos, tanto para a sugestão de duplicidades, quanto para a evolução de estórias ocorre, pois a similaridade semântica entre as estórias de usuário pode oscilar facilmente quando é encontrado qualquer relacionamento semântico entre as palavras. Por exemplo, ao considerar palavras que não sejam significativas, no que se refere à funcionalidade de uma estória de usuário, haverá uma equivocada aproximação por similaridade, e, consequentemente, um aumento no número de falsos positivos. Outra razão é que em uma base lexical, como, por exemplo, a WordNet, o número elevado de associações de palavras pode por vezes aproximar equivocadamente termos que, dentro do domínio da funcionalidade em uma estória, são distantes ou que não estão relacionados.

A análise de requisitos por similaridade semântica é uma potencial estratégia para transpor alguns desafios propostos durante a análise de requisitos textuais. As funções de

similaridade semântica, aliadas a uma base de conhecimento adicional (por exemplo, WordNet), apresentam melhores resultados que as funções as quais consideram apenas dados estatísticos vindos de contagem de ocorrência de termos coincidentes. Como as estórias de usuário são textos curtos e desejavelmente independentes, a contagem de termos coincidentes por si só não é uma estratégia suficiente para a indicação de similaridade em estórias de usuário, já que os resultados apresentados não foram animadores.

A análise por meio de similaridade semântica permite que o formato simplificado de escrita de estórias de usuário seja mantido, o que desobriga a inclusão de informação adicional às estórias para a realização de controle de duplicação e evolução. A verificação automática de similaridade semântica mostra, de uma forma rápida, como estão relacionados semanticamente os requisitos do *software*, o que auxilia os envolvidos no Scrum a analisarem os requisitos de forma agrupada. Isso permitirá, por exemplo, identificar requisitos que compartilham os mesmos dados de negócio comum.

Outra conclusão importante está relacionada ao uso da técnica de agrupamento para apoiar a definição de *sprints*. O agrupamento de texto por similaridade permite identificar contextos nos requisitos em análise, ou seja, conteúdos similares ou assuntos relacionados em um conjunto de estórias de usuário. Com base nesse grupo de estórias similares, é possível confirmar a escolha de determinadas estórias para um *sprint* proposto. Além disso, a identificação de grupos de estórias relacionadas permitiu sugerir rearranjos na composição de *sprints* a fim de otimizá-los a ponto de possuírem maior valor de negócio e serem compostos por estórias mais coesas.

9.1 Contribuições

A abordagem proposta neste trabalho para a análise de estórias de usuário visa apoiar atividades do ciclo de vida da metodologia ágil Scrum, como o *backlog grooming* e o planejamento de *sprint*. No refinamento do *backlog*, a identificação automatizada de casos de duplicidades e evolução entre estórias de usuário reduz tempo e esforço destinados à atualização do *backlog* quando ele possuir um grande número de estórias. A verificação de duplicidades em requisitos permitirá aos envolvidos no refinamento do *backlog* tomarem decisões assertivas a respeito do novo requisito, enquanto que a identificação de estórias que serão modificadas permitirá aos envolvidos levantar rapidamente os pontos do sistema que

foram implementados e que serão alterados. No planejamento dos *sprints*, os envolvidos poderão confirmar a composição de seus *sprints* por meio do método de agrupamento de grupos de estórias de usuário proposto neste trabalho.

9.2 Aplicações

O método para análise de estórias de usuário proposto neste trabalho permite o desenvolvimento de ferramentas automatizadas que visam apresentar informações úteis aos participantes do Scrum, bem como que os apoiem em decisões sobre o desenvolvimento de um *software*. O uso de agrupamento de texto e análise de similaridade semântica é uma abordagem bem sucedida que pode ser empregada em ferramentas que apoiam atividades de gerenciamento de requisitos, como, por exemplo, identificação de duplicidades, mapeamento de requisitos e seus relacionamentos, redundâncias e controle de mudanças.

Acredita-se que o método para a análise de estórias desenvolvido neste trabalho abre caminho para novas pesquisas e tem valor significativo para a comunidade científica e para a comunidade de engenharia de requisitos.

9.3 Trabalhos Futuros

O uso de agrupamento de texto e similaridade semântica para análise de requisitos textuais permitiu que fossem descobertos relacionamentos entre requisitos. Tais relacionamentos podem ser aproveitados para se estender este trabalho com o objetivo de se criar um modelo de estimação de esforço para implantação de estórias de usuário. Nesse caso, o agrupamento de estórias de usuário por similaridade poderá gerar grupos de estórias similares que compartilham níveis de esforço similares e, a partir desses grupos, poder-se-ia sugerir estimativas de esforço para as novas estórias de usuário.

Ainda como trabalhos futuros, fica a ideia de implementar melhorias no método de análise de estórias para sugestão de duplicidade e evolução a fim de diminuir o número de falsos positivos apresentados nos experimentos deste trabalho. Como melhoria, pretende-se considerar fragmentos particulares extraídos das funcionalidades das estórias, em vez de considerá-las integralmente. Acredita-se que analisar palavras com funções sintáticas

específicas, como, por exemplo, objeto direto e verbo, minimiza os ruídos causados por palavras pouco relevantes sobre os níveis de similaridade e, desse modo, reduz do número de falsos positivos.

Pretende-se também realizar novos experimentos com outras funções de similaridade, inclusive com aquelas apresentadas neste trabalho e que não revelaram bons níveis de sensibilidade e precisão nos experimentos realizados. Outra possibilidade para auxiliar a redução do número de falso positivos é a utilização da técnica de aprendizagem supervisionada. Com ela e com alguns exemplos propostos por um analista humano será possível construir um modelo para melhorar a precisão de indicação de duplicidade e evolução em estórias de usuário.

Pretende-se, futuramente, utilizar o processo de agrupamento de textos para identificar e rotular contextos em grupos de estórias de usuário. Nesse caso, planeja-se utilizar uma lista de palavras chaves que servirá de consulta para identificar estórias que, de alguma forma, estejam em um domínio específico.

No que se refere à análise de distribuição de estórias em *sprints* por meio da técnica de agrupamento de textos, deseja-se realizar novos experimentos, utilizando uma base de dados com um volume maior de estórias de usuário. Com relação ao algoritmo de sugestão de alteraçoes em *sprint*, pretende-se realizar experimentos que possam comprovar sua eficiência.

Este trabalho de mestrado originou publicações no *International Workshop on Recent Advances in the DependabIlity AssessmeNt of Complex systEms* - RADIANCE @ DSN 2015 com os título "*An Approach to Clustering and Sequencing of Textual Requirements*" e, em RADIANCE @ DSN 2016, com o título "*Use of similarity measure to suggest the existence of duplicate user stories in the Scrum process*".

Referências

Aasem, M., Ramzan, M., & Jaffar, A. (2010). Analysis and optimization of software requirements prioritization techniques. *Information and Emerging Technologies (ICIET), 2010 International Conference on*. doi:10.1109/ICIET.2010.5625687

Abrahamsson, P., Fronza, I., Moser, R., Vlasenko, J., & Pedrycz, W. (2011). Predicting development effort from user stories. In *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on* (pp. 400–403).

Al-Otaiby, T. N., AlSherif, M., & Bond, W. P. (2005). Toward Software Requirements Modularization Using Hierarchical Clustering Techniques. In *Proceedings of the 43rd Annual Southeast Regional Conference - Volume 2* (pp. 223–228). New York, NY, USA: ACM. doi:10.1145/1167253.1167305

Article Rewriter Tool. (2016). Retrieved from http://articlerewritertool.com/

Baghela, V. S., & Tripathi, S. P. (2012). Text Mining Approaches To Extract Interesting Association Rules Association Rules from Text Documents Text Documents. *International Journal of Computer Science Issues*, *9*(3).

Banerjee, A. (2011). Requirement Evolution Management: A Systematic Approach. In *2011 IEEE Computer Society Annual Symposium on VLSI* (pp. 150–155). doi:10.1109/ISVLSI.2011.20

Banerjee, S., & Pedersen, T. (2002a). An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Computational linguistics and intelligent text processing* (pp. 136–145). Springer.

Banerjee, S., & Pedersen, T. (2002b). An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. *Computational Linguistics and Intelligent Text Processing*, *2276*, 136–145. doi:10.1007/3-540-45715-1_11

Barbosa, R., Silva, A. E. A., & Moraes, R. (2016). Use of Similarity Measure to Suggest the Existence of Duplicate User Stories in the Scrum Process. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)* (pp. 2–5). doi:10.1109/DSN-W.2016.27

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). Agile Manifesto. Retrieved from http://agilemanifesto.org/

Bildhauer, D., Horn, T., & Ebert, J. (2009). Similarity-driven software reuse. In *Proceedings of the 2009 ICSE Workshop on Comparison and Versioning of Software Models, CVSM 2009* (pp. 31–36). doi:10.1109/CVSM.2009.5071719

Buglione, L., & Abran, A. (2013). Improving the user story Agile technique using the INVEST criteria. In *Proceedings - Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, IWSM-MENSURA 2013* (pp. 49–53). doi:10.1109/IWSM-Mensura.2013.18

Casamayor, A., Godoy, D., & Campo, M. (2011). Mining textual requirements to assist architectural software design: a state of the art review. *Artificial Intelligence Review*, *38*(3), 173–191. doi:10.1007/s10462-011-9237-7

Casamayor, A., Godoy, D., & Campo, M. (2012). Functional grouping of natural language requirements for assistance in architectural software design. *Knowledge-Based Systems*, *30*, 78–86. doi:10.1016/j.knosys.2011.12.009

Castro-Herrera, C., Cleland-Huang, J., & Mobasher, B. (2009). Enhancing stakeholder profiles to improve recommendations in online requirements elicitation. In *Proceedings of the IEEE International Conference on Requirements Engineering* (pp. 37–46). doi:10.1109/RE.2009.20

Castro-Herrera, C., Duan, C., Cleland-Huang, J., & Mobasher, B. (2009). A recommender system for requirements elicitation in large-scale software projects. In *Proceedings of the 2009 ACM symposium on Applied Computing - SAC '09* (pp. 1419–1426). doi:10.1145/1529282.1529601

Castro-Herrera, C., Duan, C. D. C., Cleland-Huang, J., & Mobasher, B. (2008). Using Data Mining and Recommender Systems to Facilitate Large-Scale, Open, and Inclusive Requirements Elicitation Processes. *2008 16th IEEE International Requirements Engineering Conference*. doi:10.1109/RE.2008.47

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *City*, *1*(2), 1.

Cleland-Huang, J., & Mobasher, B. (2008). Using Data Mining and Recommender Systems to Scale up the Requirements Process. *Proceedings of the 2nd International Workshop on Ultra-Large-Scale Software-Intensive Systems - ULSSIS '08*, 3–6. doi:10.1145/1370700.1370702

Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.

Cohn, M. (2008). Advantages of the "As a user, I want" user story template. Retrieved from http://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template

Cope, J. (2013). Institutional Data Repository User Stories. Bath: University of Bath. Retrieved from http://opus.bath.ac.uk/34082/

COTUCA. (2015). Retrieved October 1, 2015, from http://www.cotuca.unicamp.br

De Marneffe, M.-C., MacCartney, B., Manning, C. D., & others. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC* (Vol. 6, pp. 449–454).

Dias-da-Silva, B., de Oliveira, M., & de Moraes, H. (2002). Groundwork for the development of the Brazilian Portuguese Wordnet. In *Advances in natural language processing* (Vol. 2389, pp. 179–192). doi:10.1007/3-540-45433-0_28

Dolan, W. B., & Brockett, C. (2005). Automatically Constructing a Corpus of Sentential Paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation

of Natural Language Processing. Retrieved from http://research.microsoft.com/apps/pubs/default.aspx?id=101076

Duan, C. (2008). Clustering and its application in requirements engineering. *Technical Reports*, 7.

Duan, C., & Cleland-Huang, J. (2007a). A Clustering Technique for Early Detection of Dominant and Recessive Cross-Cutting Concerns. *Early Aspects at ICSE: Workshops in Aspect-Oriented Requirements Engineering and Architecture Design (EARLYASPECTS'07)*. doi:10.1109/EARLYASPECTS.2007.1

Duan, C., & Cleland-Huang, J. (2007b). Clustering support for automated tracing. *Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering - ASE '07*, 244. doi:10.1145/1321631.1321668

Duan, C., Laurent, P., Cleland-Huang, J., & Kwiatkowski, C. (2009). Towards automated requirements prioritization and triage. *Requirements Engineering*. doi:10.1007/s00766-009-0079-7

Ebecken, N., Lopes, M., & Costa, M. (2003). Mineração de Textos. In *Sistemas Inteligentes: Fundamentos e Aplicações* (pp. 337–370).

Embrechts, M. J., Gatti, C. J., Linton, J., & Roysam, B. (2013). Advances in Intelligent Signal Processing and Data Mining: Theory and Applications. In P. Georgieva, L. Mihaylova, & C. L. Jain (Eds.), (pp. 197–233). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-28696-4_8

Fabbrini, F., Fusani, M., Gnesi, S., & Lami, G. (2007). Controlling requirements evolution: A formal concept analysis-based approach. In *2nd International Conference on Software Engineering Advances - ICSEA 2007*. doi:10.1109/ICSEA.2007.24

Feldman, R., & Dagan, I. (1995). Knowledge Discovery in Textual Databases (KDT). *International Conference on Knowledge Discovery and Data Mining (Kdd)*, 112–117. doi:10.1.1.47.7462

Feldman, R., & Sanger, J. (2007). The text mining handbook: advanced approaches in analyzing unstructured data. *Imagine*, *34*, 410. doi:10.1179/1465312512Z.00000000017

Fellbaum, C. (1998). WordNet: An Electronic Lexical Database. *MIT Press, Cambridge, London, England*. doi:10.1139/h11-025

Ferrari, A., Gnesi, S., & Tolomei, G. (2012). A Clustering-based Approach for Discovering Flaws in Requirements Specifications. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 1043–1050). New York, NY, USA: ACM. doi:10.1145/2245276.2231939

Filippone, M., Camastra, F., Masulli, F., & Rovetta, S. (2008). A survey of kernel and spectral methods for clustering. *Pattern Recognition*, *41*, 176–190. doi:10.1016/j.patcog.2007.05.018

Francis, W. N., Kučera, H., & Mackie, A. W. (1982). *Frequency analysis of English usage: lexicon and grammar*. Houghton Mifflin. Retrieved from

https://books.google.com.br/books?id=i7AUAQAAIAAJ

Free Article Spinner. (2017). Retrieved from http://free-article-spinner.com/

Gao, J., Zhang, L., & Jiang, W. (2007). Procuring Requirements for ERP Software Based on Semantic Similarity. In *International Conference on Semantic Computing (ICSC 2007)* (pp. 61–70). doi:10.1109/ICSC.2007.45

Gervasi, V., & Zowghi, D. (2011). Mining requirements links. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 6606 LNCS, pp. 196–201). Retrieved from http://dx.doi.org/10.1007/978-3-642-19858-8_19

Gomaa, W. H., & Fahmy, A. A. (2013). Article: A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, *68*(13), 13–18.

Grapenthin, S., Poggel, S., Book, M., & Gruhn, V. (2014). Facilitating Task Breakdown in Sprint Planning Meeting 2 with an Interaction Room: An Experience Report. In *Software Engineering and Advanced Applications (SEAA), 2014 40th EUROMICRO Conference on* (pp. 1–8). doi:10.1109/SEAA.2014.71

Greghi, J. G., Martins, R. T., & das Graças Volpe Nunes, M. (2002). DIADORIM - A Lexical Database for Brazilian Portuguese. In *LREC*. European Language Resources Association. Retrieved from http://dblp.uni-trier.de/db/conf/lrec/lrec2002.html#GreghiMN02

Gregorio, D. D. (2012). How the Business Analyst supports and encourages collaboration on agile projects. In *Systems Conference (SysCon), 2012 IEEE International* (pp. 1–4). doi:10.1109/SysCon.2012.6189437

Guelpeli, M. V. C. (2012). *CASSIOPEIA Um modelo de agrupamento baseado em sumarização.* Universidade Federal Fluminense.

Güldali, B., Funke, H., Sauer, S., & Engels, G. (2011). TORC: test plan optimization by requirements clustering. *Software Quality Journal*. doi:10.1007/s11219-011-9149-4

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002a). Cluster validity methods: part I. *ACM Sigmod Record*, *31*(2), 40–45.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2002b). Clustering validity checking methods: part II. *ACM SIGMOD Record*, *31*(3), 19. doi:10.1145/601858.601862

Harman, D. (1991). How Effective Is Suffixing ? *Journal of the American Society for Information Science*, *42*(1), 7–15. doi:10.1002/(SICI)1097-4571(199101)42:1<7::AID-ASI2>3.0.CO;2-P

Hayes, J. H., Antoniol, G., & Gueheneuc, Y.-G. (2008). PREREQIR: Recovering Pre-Requirements via Cluster Analysis. *2008 15th Working Conference on Reverse Engineering*. doi:10.1109/WCRE.2008.36

Heidenberg, J., Weijola, M., Mikkonen, K., & Porres, I. (2012). A model for business value in large-scale agile and lean software development. In *European Conference on Software Process Improvement* (pp. 49–60).

Hotho, A., Nürnberger, A., & Paaß, G. (2005a). A Brief Survey of Text Mining. *Machine Learning*, *20*, 19–62. doi:10.1111/j.1365-2621.1978.tb09773.x

Hotho, A., Nürnberger, A., & Paaß, G. (2005b). A Brief Survey of Text Mining. *Ldv Forum*, 1–37. Retrieved from http://www.kde.cs.uni-kassel.de/hotho/pub/2005/hotho05TextMining.pdf

Hsia, P., & Gupta, a. (1992a). Incremental delivery using abstract data types and requirements clustering. *Proceedings of the Second International Conference on Systems Integration*, 137–150. doi:10.1109/ICSI.1992.217275

Hsia, P., & Gupta, A. (1992b). Incremental delivery using abstract data types and requirementsclustering. *Proceedings of the Second International Conference on Systems Integration*. doi:10.1109/ICSI.1992.217275

Hsia, P., Hsu, C. T., Kung, D. C., & Holder, L. B. (1996). User-centered system decomposition: Z-based requirements clustering. *Proceedings of the Second International Conference on Requirements Engineering*. doi:10.1109/ICRE.1996.491437

Hsia, P., & Yaung, A. T. (1988). Another approach to system decomposition: requirements clustering. *Proceedings COMPSAC 88: The Twelfth Annual International Computer Software & Applications Conference*. doi:10.1109/CMPSAC.1988.17153

Huang, A. (2008). Similarity measures for text document clustering. *Proceedings of the Sixth New Zealand*, 49–56. Retrieved from http://nzcsrsc08.canterbury.ac.nz/site/proceedings/Individual_Papers/pg049_Similarity_Measures_for_Text_Document_Clustering.pdf

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining,(PAKDD)*, 21–34. Retrieved from http://reference.kfupm.edu.sa/content/c/l/clustering_large_data_sets_with_mixed_nu_362883.pdf

Huang, Z. (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery*, *2*(3), 283–304. doi:10.1023/A:1009769707641

Hussain, I., Kosseim, L., & Ormandjieva, O. (2010). Towards Approximating COSMIC Functional Size from User Requirements in Agile Development Processes Using Text Mining. In *Proceedings of the Natural Language Processing and Information Systems, and 15th International Conference on Applications of Natural Language to Information Systems* (pp. 80–91). Berlin, Heidelberg: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=1894525.1894536

Hussain, I., Ormandjieva, O., & Kosseim, L. (2007). Automatic quality assessment of SRS text by means of a decision-tree-based text classifier. In *Proceedings - International Conference on Quality Software* (pp. 209–218). doi:10.1109/QSIC.2007.4385497

Hussain, Ishrar; Ormandjieva, Olga; Kosseim, L. (2009). Mining and Clustering Textual Requirements to Measure Functional Size of Software with COSMIC. In *International Conference on Software Engineering Research & Practice, SERP 2009*. Las Vegas, Nevada,

USA.

IEEE. (1998). IEEE Recommended Practice for Software Requirements SpeciÞcations. *Practice*, *1998*, 37. doi:10.1109/IEEESTD.1998.88286

Jaccard, P. (1901). Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bulletin de La Société Vaudoise Des Sciences Naturelles*, *37*, 241–272.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Jiang, J. J., & Conrath, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *Proceedings of International Conference Research on Computational Linguistics*, (Rocling X), 19–33. doi:10.1.1.269.3598

Jurafsky, D., & Martin, J. H. (2009). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. *Speech and Language Processing An Introduction to Natural Language Processing Computational Linguistics and Speech Recognition*, *21*, 0–934. doi:10.1162/089120100750105975

Kaufman, L., & Rousseeuw, P. J. (1987). Clustering by means of medoids. *Statistical Data Analysis Based on the L 1-Norm and Related Methods. First International Conference*, 405–416416.

Kaufman, L., & Rousseeuw, P. J. (1990). Finding groups in data. an introduction to cluster analysis. *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990*, *1*.

Kilgarriff, A. (2010). A detailed, accurate, extensive, available English lexical database. In *Proceedings of the NAACL HLT 2010 Demonstration Session* (pp. 21–24).

Konrad, S., & Gall, M. (2008). Requirements Engineering in the Development of Large-Scale Systems. In *International Requirements Engineering, 2008. RE '08. 16th IEEE* (pp. 217–222). doi:10.1109/RE.2008.31

Laurent, P., Cleland-Huang, J., & Duan, C. D. C. (2007). Towards Automated Requirements Triage. *15th IEEE International Requirements Engineering Conference (RE 2007)*. doi:10.1109/RE.2007.63

Leacock, C., & Chodorow, M. (1998). Combining Local Context and WordNet Similarity for Word Sense Identification. *WordNet: An Electronic Lexical Database.*, (JANUARY 1998), 265–283. doi:citeulike-article-id:1259480

Leffingwell, D. (2011). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise* (1st ed.). Addison-Wesley Professional.

Lesk, M. (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation* (pp. 24–26). New York, NY, USA: ACM. doi:10.1145/318723.318728

Li, B., & Han, L. (2013). Distance Weighted Cosine Similarity Measure for Text

Classification. In H. Yin, K. Tang, Y. Gao, F. Klawonn, M. Lee, T. Weise, … X. Yao (Eds.), *Intelligent Data Engineering and Automated Learning – IDEAL 2013 SE - 74* (Vol. 8206, pp. 611–618). Springer Berlin Heidelberg. doi:10.1007/978-3-642-41278-3_74

Li, Y., Bandar, Z. A., & McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, *15*(4), 871–882. doi:10.1109/TKDE.2003.1209005

Li, Y., McLean, D., Bandar, Z. A., O'Shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, *18*(8), 1138–1150. doi:10.1109/TKDE.2006.130

Li, Z., Rahman, Q., & Madhavji, N. (2007). An Approach to Requirements Encapsulation with Clustering. *WER*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.218.4464

Lin, D. (1998). An information-theoretic definition of similarity. In *ICML* (Vol. 98, pp. 296–304).

Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, *11*(June), 22–31. Retrieved from http://journal.mercubuana.ac.id/data/MT-1968-Lovins.pdf

Lucchese, C., Orlando, S., Perego, R., Silvestri, F., & Tolomei, G. (2011). Identifying Task-based Sessions in Search Engine Query Logs Categories and Subject Descriptors. In *WSDM* (pp. 277–286). doi:10.1145/1935826.1935875

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, *1*, 281–297. Retrieved from http://books.google.com/books?hl=en&lr=&id=IC4Ku_7dBFUC&oi=fnd&pg=PA281&dq=SOME+METHODS+FOR+CLASSIFICATION+AND+ANALYSIS+OF+MULTIVARIATE+OBSERVATIONS&ots=nMYdB0OhuL&sig=N46jBcReO1y74cX1CtHG6qgiB8Q

Mahgoub, H. (2006). Mining Association Rules from Unstructured Documents. *Enformatika*, *14*.

Manifesto Ágil. (2015). Retrieved March 1, 2015, from http://www.agilemanifesto.org/

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval Introduction. Computational Linguistics* (Vol. 35). doi:10.1162/coli.2009.35.2.307

Matsuoka, J., & Lepage, Y. (2011). Ambiguity spotting using wordnet semantic similarity in support to recommended practice for Software Requirements Specifications. *2011 7th International Conference on Natural Language Processing and Knowledge Engineering*, 479–484. doi:10.1109/NLPKE.2011.6138247

McFadden, R. R., & Mitropoulos, F. J. (2012). Aspect mining using model-based clustering. In *Southeastcon, 2012 Proceedings of IEEE* (pp. 1–8). doi:10.1109/SECon.2012.6196984

Meng, L., Huang, R., & Gu, J. (2013). A review of semantic similarity measures in wordnet. *International Journal of Hybrid Information Technology*, *6*(1), 1–12.

Meng, L., Huang, R., & Gu, J. (2014). Measuring Semantic Similarity of Word Pairs Using Path and Information Content 1. *International Journal of Future Generation Communication and Networking*, *7*(3), 183–194. doi:10.14257/ijfgcn.2014.7.3.17

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Commun. ACM*, *38*(11), 39–41. doi:10.1145/219717.219748

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, *50*(2), 159–179. doi:10.1007/BF02294245

Moldovan, G. S., & Serban, G. (2006). Aspect Mining using a Vector-Space Model Based Clustering Approach. In *Proceedings of Linking Aspect Technology and Evolution (LATE) Workshop* (pp. 36–40). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.1810&rep=rep1&type=pdf

Natt och Dag, J., Regnell, B., Carlshamre, P., Andersson, M., & Karlsson, J. (2002). A Feasibility Study of Automated Natural Language Requirements Analysis in Market-Driven Development. *Requirements Engineering*, *7*, 20–33. doi:10.1007/s007660200002

Niu, N., & Mahmoud, A. (2012). Enhancing candidate link generation for requirements tracing: The cluster hypothesis revisited. In *Requirements Engineering Conference (RE), 2012 20th IEEE International* (pp. 81–90). doi:10.1109/RE.2012.6345842

Palmer, J. D., & Liang, Y. (1992). Indexing and clustering of software requirements specifications. *Information and Decision Technologies*, *18*(4), 283–299.

Paraphrasing Tool. (2015). Retrieved October 25, 2015, from http://paraphrasing-tool.com/

Patton, J. (2008). The new user story backlog is a map. Retrieved from http://www.agileproductdesign.com/blog/the_new_backlog.html

Patton, J. (2011). Telling Better Stories with User Story Mapping. In *Agile 2011*. Retrieved from http://program2011.agilealliance.org/event/c2a0f36a9a91c82e376543c1e8f9d684\nhttp://www.agileproductdesign.com/presentations/user_story_mapping/index.html

Pedersen, T. (2010). Information content measures of semantic similarity perform better without sense-tagged text. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 329–332).

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet::Similarity: Measuring the Relatedness of Concepts. In *Demonstration Papers at HLT-NAACL 2004* (pp. 38–41). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1614025.1614037

Pelleg, D., & Moore, A. W. (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning table of contents* (pp. 727–734). Retrieved from http://portal.acm.org/citation.cfm?id=657808

Pham, A., & Van Pham, P. (2011). *Scrum in Action: Agile Software Project Management and Development*. Course Technology. Retrieved from http://books.google.com.br/books?id=QTwVkgAACAAJ

Pich., C. (2009). MDSJ: Java Library for Multidimensional Scaling (Version 0.2). Retrieved June 13, 2015, from http://www.inf.uni-konstanz.de/algo/software/mdsj/

Port, D., Nikora, A., Hayes, J. H., & Huang, L. (2011). Text Mining Support for Software Requirements: Traceability Assurance. *2011 44th Hawaii International Conference on System Sciences*, 1–11. doi:10.1109/HICSS.2011.399

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*. doi:10.1108/eb046814

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man and Cybernetics*, *19*(1), 17–30. doi:10.1109/21.24528

Ramos, J., Eden, J., & Edu, R. (2003). Using TF-IDF to Determine Word Relevance in Document Queries. *Processing*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.1424&rep=rep1&type=pdf

Resnik, P. (1995). Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *Roceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1 - IJCAI'95*, *1*, 6. doi:10.1.1.55.5277

Rezende, S. O., Marcacini, R. M., & Moura, M. F. (2011). O uso da Mineração de Textos para Extração e Organização Não Supervisionada de Conhecimento. *Revista de Sistemas de Informação Da FSMA*, *7*, 7–21.

Rijsbergen, C. J. Van. (1979). *Information Retrieval* (2nd ed.). Newton, MA, USA: Butterworth-Heinemann.

Rosenhainer, L. (2004). Identifying Crosscutting Concerns in Requirements Specifications. In *Early Aspects Workshop 2004: Aspect-Oriented Requirements Engineering and Architecture Design, OOPSLA 2004* (pp. 49–58).

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*. doi:10.1016/0377-0427(87)90125-7

Rus, V., Lintean, M. C., Banjade, R., Niraula, N. B., & Stefanescu, D. (2013). SEMILAR: The Semantic Similarity Toolkit. In *ACL (Conference System Demonstrations)* (pp. 163–168).

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*, 613–620. doi:10.1145/361219.361220

Sampaio, A., Loughran, N., Rashid, A., & Rayson, P. (2005). Mining Aspects in Requirements. *Early Aspects 2005: Aspect-Oriented Requirements Engineering and Architecture Design Workshop, 2005-03-15, Chicago, Illinois, USA.*

Sateli, B., Angius, E., Rajivelu, S. S., & Witte, R. (2012). Can Text Mining Assistants Help to Improve Requirements Specifications? In *Mining Unstructured Data (MUD 2012)*. Kingston, Ontario, Canada. Retrieved from http://sailhome.cs.queensu.ca/mud/res/sateli-mud2012.pdf

Schwaber, K. (1997). Scrum development process. In *Business Object Design and Implementation* (pp. 117–134). Springer.

Scrum Alliance (2015). Product Backlog Example. Retrieved March 15, 2015, from http://www.mountaingoatsoftware.com/agile/scrum/product-backlog/example

Scrum Alliance. (2016). Retrieved February 25, 2016, from https://www.scrumalliance.org/

Serban, G. S. G., & Moldovan, G. S. M. G. S. (2006). A New k-means Based Clustering Algorithm in Aspect Mining. *2006 Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*. doi:10.1109/SYNASC.2006.5

Slimani, T. (2013). Description and Evaluation of Semantic Similarity Measures Approaches. *International Journal of Computer Applications*, *80*(10), 25–33. doi:10.5120/13897-1851

Sommerville, I. (2011). *Software Engineering* (Ninth Edit.).

Song, S., & Li, C. (2006). SOFSEM 2006: Theory and Practice of Computer Science: 32nd Conference on Current Trends in Theory and Practice of Computer Science, Merin, Czech Republic, January 21-27, 2006. Proceedings. In J. Wiedermann, G. Tel, J. Pokorný, M. Bieliková, & J. Štuller (Eds.), (pp. 501–510). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/11611257_48

Stettina, C. J., & Heijstek, W. (2011). Necessary and neglected?: an empirical study of internal documentation in agile software development teams. In *Proceedings of the 29th ACM international conference on Design of communication* (pp. 159–166). doi:10.1145/2038476.2038509

Sutherland, J. (2004). Agile development: Lessons learned from the first scrum. *Cutter Agile Project Management Advisory Service: Executive Update*, *5*(20), 1–4.

Sutherland, J., & Schwaber, K. (2007). The Scrum Papers : Nuts , Bolts , and Origins of an Agile Process. *Origins*, (December), 1–202.

Takeuchi, H., & Nonaka, I. (1986). The New New Product Development Game. doi:10.1016/0737-6782(86)90053-6

Turk, D., France, R., & Rumpe, B. (2005). Assumptions Underlying Agile Software Development Processes. *Journal of Database Management*, *16*, 62–87. doi:10.4018/jdm.2005100104

Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E. G. M., & Milios, E. E. (2005). Semantic similarity methods in wordNet and their application to information retrieval on the web. *Proceedings of the Seventh ACM International Workshop on Web Information and Data Management - WIDM '05*, 10. doi:10.1145/1097047.1097051

Vendramin, L., Campello, R. J. G. B., & Hruschka, E. R. (2010). Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, *3*(4), 209–235.

doi:10.1002/sam.10080

Viklund, A. (2005). JFreeChart. Retrieved April 25, 2015, from http://www.jfree.org/jfreechart/

Wake, B. (2003). INVEST in Good Stories, and SMART Tasks. Retrieved April 25, 2015, from http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/

Womack, J. P., Jones, D. T., & Roos, D. (1990). *The Machine that Changed the World: The Story of Lean Production*. *World*. doi:10.1016/0024-6301(92)90400-V

Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* (pp. 133–138).

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*. doi:10.1109/TNN.2005.845141

Yale-Loehr, A., Schlesinger, I. D., Rembert, A. J., & Blake, M. B. (2010). Discovering shared services from cross-organizational software specifications. In *Proceedings - 2010 IEEE 7th International Conference on Services Computing, SCC 2010* (pp. 186–193). doi:10.1109/SCC.2010.68

Yang, Y. (1999). An Evaluation of Statistical Approaches to Text Categorization. *Inf. Retr.*, *1*(1-2), 69–90. doi:10.1023/A:1009982220290

Apêndice A

Estórias De Usuário da base SAWeb (os sublinhados indicam os trechos utilizados nos experimentos)

0 - As a site member I want to describe myself on my own page in a semi-structured way
1 - As a site member I can fill out an application to become a Practitioner
2 - As a Practitioner I want my profile page to include additional details about me
3 - As a site member I can fill out an application to become a Trainer
4 - As a Trainer I want my profile page to include additional details about me
5 - As a Practitioner or Trainer I want a small graphic associated with the content indicating I am a Practitioner or Trainer
6 - As a trainer I want my profile to list my upcoming classes and include a link to a detailed page about each
7 - As a site member I can view the profiles of other members
8 - As a site member I can search for profiles based on a few fields
9 - As a site member I can mark my profile as private in which case only my name will appear
10 - As a site member I can mark my email address as private even if the rest of my profile is not
11 - As a site member I can send an email to any member via a form
12 - As a site administrator I can read practicing and training applications and approve or reject them
13 - As a site administrator I can edit any site member profile
14 - As a site visitor I can read current news on the home page
15 - As a site visitor I can access old news that is no longer on the home page
16 - As a site visitor I can email news items to the editor
17 - As site editor I can set the following dates on a news item: Start Publishing Date, Old News Date, Stop Publishing Date
18 - As a site member I can subscribe to an RSS feed of news
19 - As a site editor I can assign priority numbers to news items
20 - As a site visitor I can see a list of all upcoming Certification Courses
21 - As a site visitor I can see a list of all upcoming Other Courses
22 - As a site visitor I can see a list of all upcoming Events
23 - As a trainer I can create a new course or event
24 - As a trainer I am charged a listing fee for that activity
25 - As a site administrator I can create an Other Course or Event that is not charged a listing fee so that the Scrum Alliance doesn't charge itself for Scrum Gatherings that it puts on
26 - As a site administrator I can set the listing fee per Other Course or Event

27 - As a trainer I can update one of my existing courses or events
28 - As a trainer I can delete one of my courses or events
29 - As a trainer I can copy one of my courses or events so that I can create a new one
30 - As a site admin I can delete any course or event
31 - As a site editor I can update any course or event
32 - As a trainer, admin, or editor, I can turn a course into an event or an event into a course
33 - As a site visitor I have an advanced search option that lets me fill in a form of search criteria
34 - As a site visitor I can click on the name of trainer and be taken to the profile of trainer
35 - As a site visitor I can subscribe to an RSS feed of upcoming courses and events
36 - As a site visitor I can read FAQs
37 - As a site editor I can maintain an FAQ section
38 - As a site member I can do a full-text search of the FAQs
39 - As a site member I can download the latest training material and methodology PDFs
40 - As a visitor I can download presentations, PDFs, etc. on Scrum that I can use
41 - As a site member I can scroll through a listing of jobs
42 - As someone who wants to hire I can post a help wanted ad
43 - As a site admin I need to approve each help wanted ad before it gets to the site
44 - As a site admin I am emailed whenever a job is submitted
45 - As a site member I can subscribe to an RSS feed of jobs available
46 - As a site admin I can edit and delete help wanted ads
47 - As a site admin I want jobs to stop publishing on the site 30 days after being posted
48 - As someone who wants to hire I want to be able to extend an ad for another 30 days by visiting the site and updating the posting
49 - As someone who has posted an ad that is about to expire, seven days before it expires I want to be emailed a reminder so that I can go extend the ad
50 - As a site visitor I want to read a new article on the front page about once a week
51 - As the site editor I can include a teaser with each article
52 - As a site member who has read a teaser on the front page I want to read the entire article
53 - As the site editor I can add an article to the site
54 - As a site editor I can set start publishing dates, old article date and stop publishing dates for articles
55 - As a site editor I want to be able to designate whether or not an article ever makes the home page
56 - As the site editor I have pretty good control over how the article looks
57 - As a site visitor I want the link from the article teaser to take me directly to the body of the article
58 - As a site editor I want to be able to indicate whether an article is publicly available or for members only
59 - As a site visitor I want to be able to read some of your articles
60 - As a site member I want to have full access to all articles

61 - As a site visitor I can do a full-text search of article body, title, and author name
62 - As a site visitor I can subscribe to an RSS feed of articles
63 - As a site visitor I can post comments about articles so that others can read them
64 - As a site editor I want to have a prominent area on the home page where I can put special announcements, not necessarily news or articles
65 - As a site editor I would like to have some flexibility as to where things appear to accommodate different types of content
66 - As a site member I want notify the visitors about upcoming courses
67 - As a site visitor I want to see new content when I come to the site
68 - As a site visitor I want to have articles that interest me and are easy to get to
69 - As a site editor I have ideas on how I want the home page to look and feel
70 - As a site visitor I need to know as soon as I visit what on earth Scrum is, and why it needs an alliance
71 - As a site visitor I want to know as I glance around the home page what on earth a CSM is and why I would want to be one
72 - As a site visitor I want to be able to get back to the home page quickly and easily
73 - As a site visitor I want to see a list of the most popular items on the site
74 - As someone who successfully completed a Certification Course I am emailed a link to a survey about the course and instructor
75 - As a trainer I want to be assured that no one can submit the same answers multiple time and skew my results
76 - As a trainer I am notified about the results of surveys about my classes
77 - As a site admin I can see the results for each trainer and averages for the class
78 - As a site visitor who is considering attending a certification course I want to see a trainer's rating
79 - As a trainer I want my rating to show up on my profile page
80 - As a site visitor I want there to be a section of the website that teaches me the basics of what Scrum is
81 - As a site editor I can create the content of the What Is Scrum section
82 - As a site visitor I can view lists on the site of all Certified Scrum Masters, Practitioners, Trainers, and Certified *Product* Owners
83 - As a CSM, Practitioner, or Certified *Product* Owner I can have my name listed in the registry without becoming a member of the site
84 - As a trainer who has finished teaching a Certification class I can load an Excel file into the site
85 - As a site admin I can view all classes in a pending state
86 - As a site admin who has received proof of payment from a trainer I can move people in his or her class from a pending state to the registry
87 - As a new Certified Scrum Master or Certified *Product* Owner, once my name has been loaded to the registry I am sent an email welcoming me to the Scrum Alliance and containing instructions on how to register / activate my membership
88 - As a site editor I can edit the content of the email automatically sent to new Certified Scrum Masters and *Product* Owners
89 - As a company I can join the Scrum Alliance by paying a corporate membership fee. This will include uploading items related to corporate membership
90 - As a corporate sponsor I want my logo is displayed on a Corporate Sponsors page
91 - As a corporate sponsor I want my logo to randomly appear on the home page
92 - As a Certified Scrum Master or Certified *Product* Owner who has been approved for Practitioner status, I am charged a fee
93 - As someone about to become a trainer I can pay an annual fee

94 - As a site administrator I can set the annual fees for members, Practitioners and Trainers
95 - As someone whose membership is about to expire I am sent a reminder and a link through which I can renew
96 - As a member with short-term memory problems I can have the system email me a new password or a password reminder and son on
97 - As a trainer I can read information of relevance only to trainers
98 - As a site editor I can post information in a trainers-only section

Estórias de usuário da base RepBath (os sublinhados indicam os trechos utilizados nos experimentos)

0 - As a depositor I want to Deposit and maintain datasets through a simple web interface so that I don't need to install and learn new *software* to deposit
1 - As a depositor I want to Have a user interface that is familiar to me so that I feel like all the University systems are joined up
2 - As a depositor I want to Deposit and maintain datasets through Pure so that I have a single one-stop shop for managing my research outputs
3 - As a depositor I want to Deposit and maintain datasets through Virtual Research Environments and other workflow tools so that I can continue to work with tools with which I'm familiar
4 - As a depositor I want to Deposit the files that I have so that I don't have to spend a lot of time finding the right version and converting to the right format
5 - As a depositor I want to Place data under an embargo so that My right of first-use is protected. I can fulfill my confidentiality responsibilities
6 - As a depositor I want to Apply licenses to datasets so that My IP rights are protected appropriately
7 - As a depositor I want to Allow my collaborators privileged access to datasets so that We continue to have a productive relationship
8 - As a depositor I want to Deposit arbitrarily large files so that I am not limited in what files I can and cannot deposit
9 - As a depositor I want to Link datasets to publications in Opus so that Both my data and publications are more easily discovered
10 - As a depositor I want to Mint DOIs for my data so that It can be discovered and cited more easily. Citations can be tracked so that I can receive credit
11 - As a depositor I want to Have metadata automatically filled from other University systems (e.g. Pure) and/or remembered from previous deposits so that I don't have to waste time re-entering the same information
12 - As a depositor I want to Link to data stored in external repositories so that I can store my data in an appropriate repository but still register it with the University. I don't have to deposit my data in multiple places
13 - As a depositor I want to Specify a retention/disposal policy for my data so that I do not accidentally breach laws or collaboration agreements
14 - As a depositor I want to Track downloads of my data so that I can demonstrate the impact of my work
15 - As a depositor I want to Track citations of my data so that I can demonstrate the impact of my work
16 - As a depositor I want to Have guarantees about data integrity so that I can use my data in the future. I can fulfill funder requirements for archival
17 - As a depositor I want to Attach subject-specific discoverability metadata to records so that Researchers in my discipline can find my data more easily
18 - As a depositor I want to Link datasets with the project DMP (possibly from DMP online) so that Compliance with DMP can be demonstrated. Whole project workflow is linked together
19 - As a depositor I want to Manage and share "live" research data so that Whole project workflow is linked together
20 - As a depositor I want to Manage multiple versions of the same dataset so that Changes to the dataset are transparent and do not compromise research integrity

21 - As a depositor I want to Allow others to deposit on my behalf so that I can delegate research data management tasks appropriately
22 - As a data re-user I want to Search the archive through the web so that I can easily find data relevant to my needs
23 - As a data re-user I want to Access the system in my native language so that I am not put off re-using University of Bath data by language barriers
24 - As a data re-user I want to Examine and identify deposited files so that I can make a preliminary assessment of usefulness without downloading the whole dataset
25 - As a data re-user I want to View an example citation for a dataset so that I can reference it correctly
26 - As a data re-user I want to View a DOI for a dataset so that I can get back to the data in future. I can import the dataset into my reference-management *software* automatically
27 - As a data re-user I want to Get a persistent URL for a dataset so that I can get back to the data in future
28 - As a data re-user I want to Search the archive through Primo (University of Bath library search system) so that I can search books, articles and data all in one place
29 - As a data re-user I want to See different versions (including the latest) of a dataset at a glance so that I can be sure I'm using the right version of the dataset
30 - As an external collaborator I want to Gain privileged access to data for projects in which I am involved so that I can collaborate effectively
31 - As an external collaborator I want to Have guarantees that my IP rights will not be breached so that The risk of collaborating with Bath is acceptable to me
32 - As an external collaborator I want to Access data from Bath collaborators off campus so that I can collaborate effectively
33 - As a research facility manager I want to Deposit data from my facility directly into the archive on behalf of researchers so that I am no longer required to maintain my own archive of facility data. Researchers can access their own data as needed
34 - As a Bath Data Archive administrator I want to Make some checks on deposited datasets before they are made public so that Consistent quality of metadata is maintained. Compliance with policies can be checked. Details of licensing can be checked
35 - As a Bath Data Archive administrator I want to Require a minimum set of metadata so that Consistent quality of metadata is maintained
36 - As a Bath Data Archive administrator I want to Approve scheduled disposal of data so that Data which is still required is not destroyed
37 - As a Bath Data Archive administrator I want to Query the entire archive (including embargoed records) so that I can report on particular aspects of the archive holdings
38 - As a Bath Data Archive administrator I want to Import Bath data from an external data centre wholesale so that Bath data holdings in external archives are not lost if they close down
39 - As a Bath Data Archive administrator I want to Encourage and promote the use of open standards for deposit so that Data is as reusable as possible
40 - As a Research Information manager I want to Integrate the archive with CRIS so that I can analyse impact of research data publication. I can link funding to all of the outputs it produces
41 - As a Research Information manager I want to Include records for externally-held data so that The university's record of data holdings is complete
42 - As a Research Information manager I want to Track citation counts for published datasets so that Impact of datasets within academia can be demonstrated
43 - As a Research Information manager I want to Segment view & download statistics by country and sector so that Impact of datasets outside academia can be demonstrated
44 - As a Research Information manager I want to Have datasets linked to metadata about projects so that I can report on projects depositing datasets in relation to funder requirements
45 - As the university IT service I want to Store archived data on existing storage systems so that University data storage is consistent and maintainable. Future availability of data can be guaranteed
46 - As the university IT service I want to Integrate the archive with existing university systems such as LDAP so that The cost of administering the system can be kept low
47 - As the university IT service I want to Store archived data directly on the HCP object store so that Best use of the HCP's features can be made

48 - As the university IT service I want to Be able to export all data to a different system so that I am not tied into one system which may not be the most appropriate at some point in the future
49 - As a developer/maintainer of related services I want to Deposit and maintain datasets via an API such as SWORD2 so that My service can interact with the archive
50 - As an academic publisher I want to Make persistent web links between my articles and underlying datasets so that My journals can be seen to be filled with robust, high-quality research
51 - As a funding body I want to Be reassured (by individual researchers or an institution) that researchers I fund have robust archival plans for their data so that I can be sure that funding them is a worthwhile investment
52 - As a funding body I want to Harvest metadata on outputs from research I fund via e.g. OAI-PMH so that I can analyse effectiveness of funding strategy. I can encourage cross-fertilisation of research outputs

Estórias de usuário da base COTUCA (os sublinhados indicam os trechos utilizados nos experimentos)

0 - As a director I want to send emails to students so that I can inform them about important news and press releases
1 - As a director I want to send the report card of students by e-mail to parents so that parents are informed of the scores of children
2 - As an associate director I want to send messages to students so that I can inform them about important matters
3 - As an associate director I want to send files to allow students to download
4 - As an associate director I want to send files to allow teachers to download
5 - As an internship coordinator I want to send emails to companies so that I can report events that happen at school
6 - As an internship coordinator I want to send e-mails to students so that I can disclose apprenticeship positions that were opened by companies
7 - As an internship coordinator I want to follow the training contracts so that I facilitate the partnership between students and businesses
8 - As an internship coordinator I want to sign new training contracts
9 - As an internship coordinator I want to query the data of companies and signed agreements for decision making
10 - As an internship coordinator I want to make changes in training contracts
11 - As an internship coordinator I want to export training contracts in PDF format to access them anywhere
12 - As an internship coordinator I want to receive the internship reports in PDF format to easily distribute them to teachers
13 - As a secretary I want to send announcements and news for students so that I can contact them quickly
14 - As a secretary I want to generate password to hand out to students
15 - As a secretary I want to generate password to hand out to parents
16 - As a secretary I want to see students who delivered stages reports to draft conclusion diplomas
17 - As a secretary I want to send announcements to the teachers so that I can contact them quickly
18 - As a student I want to consult my grades so that I do not go to school

19 - As a student I want to receive notices and news so that I do not go to school
20 - As a student I want to enrolling in courses so that I do not go to school
21 - As a student I want to enrolling in internship discipline so that I do not go to school
22 - As a student I want to send my internship report
23 - As a student I want to change my password
24 - As a student I want to see my transcript
25 - As a student I want to consult the subjects that still need to attend
26 - As a student I want to print my transcript so that I do not go to the office
27 - As a student I want to print registration certificate so that I can present it in places that require
28 - As a student I want to cancel my enrollment in any discipline so that I need not go to school
29 - As a student I want to see the result application for registration for courses
30 - As a student I want to see my GPA
31 - As a student I want to consult my internship contract
32 - As a student I want to sign my internship contract electronically so that I do not print the contract
33 - As a student I want to change my registration data
34 - As a student I want to check the timetables of the subjects of my course and other courses so that I can choose the subjects they want to attend
35 - As a student I want to print a registration achievement of proof so that I can present it in places that require
36 - As a student I want to consult the course curriculum so that I am enrolled for the courses and prerequisites
37 - As a student I want to downloading of common use of documents
38 - As a student I want to downloading of didactic material of the course so that I can study at home
39 - As a student I want to export my transcript in PDF format so that I can store it and send it with ease
40 - As a student I want to export my registration certificate in PDF format so that I can store it and send it with ease
41 - As a student I want to request a second copy of student card so that I do not go to the office
42 - As a student I want to request documents and services to the office so that I do not go to the office
43 - As a student I want to receive warnings about internship opportunities so that I do not have to consult posters
44 - As a teacher I want inform students' grades so that students can refer to them
45 - As a teacher I want consult some data from my students so that I can plan lessons
46 - As a teacher I want record the school daily disciplines that teach
47 - As a teacher I want check students' grades in other subjects so that I may see the overall student achievement
48 - As a teacher I want change my password so that I do not go to school
49 - As a teacher I want export grades of students in CSV or XLS format so that I can work with spreadsheets in Excel
50 - As a teacher I want send e-mail to students so that I can communicate students on a particular subject
51 - As a teacher I want send e-mail to the student's parents so that I can inform parents some important matter
52 - As a teacher I want provide educational material for students

53 - As a teacher I want print the school diary of my subjects so that I do not request the secretariat
54 - As a teacher I want export electronic journals in PDF so that I can store them and send them easily
55 - As a teacher I want downloading of photographs of students so that I can identify students with ease
56 - As a teacher I want deliver my school daily on the internet so that I did not have to go to school
57 - As a teacher I want download the internship reports so that I can evaluate them
58 - As a teacher I want deliver final grades via the Internet so that students can see the notes quickly
59 - As a teacher I want partial delivery notes over the Internet so that students can see the notes quickly
60 - As a teacher I want download the lesson plans in PDF format so that I can store them and send them easily
61 - As a teacher I want registering individual assessment of student information so that I can notify you of your income in a discipline
62 - As a teacher I want consult the frequency of students in remedial classes
63 - As a teacher I want consult statistics of the notes in a class so that I view the overall performance of the class
64 - As a teacher I want download commonly used files of teachers so that I do not have to ask by e-mail
65 - As a teacher I want make classroom reservation so that I do not make a reservation by e-mail
66 - As a librarian I want send announcements to the students so that I can inform them about library issues
67 - As a librarian I want send announcements to the students so that I can inform them of pending
68 - As a parent of a student I want receive the report card by email so that I do not go to school
69 - As a parent of a student I want see messages or warnings given by the teacher to me to be informed
70 - As a parent of a student I want enable or disable the receipt of communications by e-mail so that I do not receive unwanted emails

Apêndice B

Funcionalidades duplicadas geradas pela ferramenta Paraphrasing-Tool para a base de estórias SAWeb (funcionalidades sublinhadas foram as escolhidas aleatoriamente para o experimento)

0 - I need to depict myself all alone page in a semi-organized manner
1 - I can round out an application to end up a Practitioner
2 - I need my profile page to incorporate extra insights about me
3 - I can round out an application to end up a Trainer
4 - I need my profile page to incorporate extra insights about me
5 - I need a little realistic connected with the substance demonstrating I am a Practitioner or Trainer
6 - I need my profile to list my up and coming classes and incorporate a connection to a nitty gritty page about each
7 - I can see the profiles of different individuals
8 - I can hunt down profiles in view of a couple fields
9 - I can check my profile as private in which case just my name will show up
10 - I can check my email address as private regardless of the fact that whatever remains of my profile is definitely not
11 - I can send an email to any part by means of a structure
12 - I can read honing and preparing applications and support or reject them
13 - I can alter any site part profile
14 - I can read current news on the landing page
15 - I can get to old news that is no more on the landing page
16 - I can email news things to the manager
17 - I can set the accompanying dates on a news thing: Start Publishing Date, Old News Date, Stop Publishing Date
18 - I can subscribe to a RSS channel of news
19 - I can appoint need numbers to news things
20 - I can see a rundown of all forthcoming Certification Courses
21 - I can see a rundown of all forthcoming Other Courses
22 - I can see a rundown of every single forthcoming Event
23 - I can make another course or occasion
24 - I am charged a posting expense for that movement
25 - I can make an Other Course or Event that is not charged a posting expense so that the Scrum Alliance doesn't charge itself for Scrum Gatherings that it puts on

26 - I can set the posting charge per Other Course or Event
27 - I can redesign one of my current courses or occasions
28 - I can erase one of my courses or occasions
29 - I can duplicate one of my courses or occasions so I can make another one
30 - I can erase any course or occasion
31 - I can upgrade any course or occasion
32 - I can transform a course into an occasion or an occasion into a course
33 - I have a propelled seek alternative that gives me a chance to fill in a type of inquiry criteria
34 - I can tap on the name of coach and be taken to the profile of mentor
35 - I can subscribe to a RSS channel of up and coming courses and occasions
36 - I can read FAQs
37 - I can keep up a FAQ segment
38 - I can do a full-content pursuit of the FAQs
39 - I can download the most recent preparing material and technique PDFs
40 - I can download presentations, PDFs, and so on Scrum that I can utilize
41 - I can look through a posting of employments
42 - I can post a help needed commercial
43 - I have to favor every offer needed promotion before it some assistance with getting to the site
44 - I am messaged at whatever point a vocation is submitted
45 - I can subscribe to a RSS channel of occupations accessible
46 - I can alter and erase help needed promotions
47 - I need occupations to quit distributed on the site 30 days subsequent to being posted
48 - I need to have the capacity to develop a notice for an additional 30 days by going to the site and upgrading the posting
49 - I need to be messaged an update with the goal that I can go develop the notice
50 - I need to peruse another article on the front page about once every week
51 - I can incorporate a teaser with every article
52 - I need to peruse the whole article
53 - I can add an article to the site
54 - I can set begin distributed dates, old article date and quit distributed dates for articles
55 - I need to have the capacity to assign regardless of whether an article ever makes the landing page
56 - I have really great control over how the article looks
57 - I need the connection from the article teaser to take me straightforwardly to the body of the article
58 - I need to have the capacity to demonstrate whether an article is freely accessible or for individuals just
59 - I need to have the capacity to peruse some of your articles

60 - I need to have full access to all articles
61 - I can do a full-content hunt of article body, title, and writer name
62 - I can subscribe to a RSS channel of articles
63 - I can post remarks about articles so others can read them
64 - I need to have a conspicuous range on the landing page where I can put exceptional declarations, not as a matter of course news or articles
65 - I might want to have some adaptability as to where things seem to oblige diverse sorts of substance
66 - I need inform the guests about forthcoming courses
67 - I need to see new substance when I go to the site
68 - I need to have articles that intrigue me and are anything but difficult to get to
69 - I have thoughts on how I need the landing page to look and feel
70 - I have to know when I visit what on earth Scrum is, and why it needs a union
71 - I need to know as I look around the landing page what on earth a CSM
72 - I need to have the capacity to return to the landing page rapidly and effectively
73 - I need to see a rundown of the most prominent things on the site
74 - I am messaged a connection to an overview about the course and teacher
75 - I need to be guaranteed that nobody can present the same answers various time and skew my outcomes
76 - I am informed about the after effects of overviews about my classes
77 - I can see the outcomes for every coach and midpoints for the class
78 - I need to see a coach's evaluating
79 - I need my rating to appear on my profile page
80 - I need there to be a segment of the site that shows me the essentials of what Scrum is
81 - I can make the substance of the What Is Scrum segment
82 - I can view records on the site of all Certified Scrum Masters, Practitioners, Trainers, and Certified *Product* Owners
83 - I can have my name recorded in the registry without turning into an individual from the site
84 - I can stack an Excel document into the site
85 - I can see all classes in a pending state
86 - I can move individuals in his or her class from a pending state to the registry
87 - I am sent an email inviting me to the Scrum Alliance and containing directions on the most proficient method to enlist/actuate my enrollment
88 - I can alter the substance of the email naturally sent to new Certified Scrum Masters and *Product* Owners
89 - I can join the Scrum Alliance by paying a corporate enrollment expense
90 - I need my logo is shown on a Corporate Sponsors page
91 - I need my logo to arbitrarily show up on the landing page
92 - I am charged an expense
93 - I can pay a yearly expense

94 - I can set the yearly expenses for individuals, Practitioners and Trainers
95 - I am sent an update and a connection through which I can reestablish
96 - I can have the framework email me another secret key or a watchword update and child on
97 - I can read data of significance just to mentors
98 - I can post data in a mentors just area

Funcionalidades duplicadas geradas pela ferramenta Free Article Spinner para a base de estórias SAWeb (funcionalidades sublinhadas foram as escolhidas aleatoriamente para o experimento)

0 - I need to depict myself all alone page in a semi-organized manner
1 - I can round out an application to wind up a Practitioner
2 - I need my profile page to incorporate extra insights about me
3 - I can round out an application to wind up a Trainer
4 - I need my profile page to incorporate extra insights about me
5 - I need a little realistic connected with the substance showing I am a Practitioner or Trainer
6 - I need my profile to list my forthcoming classes and incorporate a connection to a point by point page about each
7 - I can see the profiles of different individuals
8 - I can scan for profiles in light of a couple fields
9 - I can check my profile as private in which case just my name will show up
10 - I can stamp my email address as private regardless of the fact that whatever is left of my profile is most certainly not
11 - I can send an email to any part by means of a structure
12 - I can read rehearsing and preparing applications and endorse or dismiss them
13 - I can alter any site part profile
14 - I can read current news on the landing page
15 - I can get to old news that is no more on the landing page
16 - I can email news things to the manager
17 - I can set the accompanying dates on a news thing: Start Publishing Date, Old News Date, Stop Publishing Date
18 - I can subscribe to a RSS channel of news
19 - I can relegate need numbers to news things
20 - I can see a rundown of all forthcoming Certification Courses
21 - I can see a rundown of all forthcoming Other Courses

22 - I can see a rundown of every single forthcoming Event
23 - I can make another course or occasion
24 - I am charged a posting expense for that action
25 - I can make an Other Course or Event that is not charged a posting expense so that the Scrum Alliance doesn't charge itself for Scrum Gatherings that it puts on
26 - I can set the posting expense per Other Course or Event
27 - I can upgrade one of my current courses or occasions
28 - I can erase one of my courses or occasions
29 - I can duplicate one of my courses or occasions with the goal that I can make another one
30 - I can erase any course or occasion
31 - I can redesign any course or occasion
32 - I can transform a course into an occasion or an occasion into a course
33 - I have a propelled look choice that gives me a chance to fill in a type of pursuit criteria
34 - I can tap on the name of coach and be taken to the profile of mentor
35 - I can subscribe to a RSS channel of up and coming courses and occasions
36 - I can read FAQs
37 - I can keep up a FAQ segment
38 - I can do a full-message inquiry of the FAQs
39 - I can download the most recent preparing material and procedure PDFs
40 - I can download presentations, PDFs, and so forth on Scrum that I can utilize
41 - I can look through a posting of occupations
42 - I can post a help needed promotion
43 - I have to favor every help needed advertisement before it gets to the site
44 - I am messaged at whatever point a vocation is submitted
45 - I can subscribe to a RSS channel of occupations accessible
46 - I can alter and erase help needed promotions
47 - I need employments to quit distributed on the site 30 days subsequent to being posted
48 - I need to have the capacity to broaden an advertisement for an additional 30 days by going to the site and upgrading the posting
49 - I need to be messaged an update with the goal that I can go augment the advertisement
50 - I need to peruse another article on the front page about once per week
51 - I can incorporate a teaser with every article
52 - I need to peruse the whole article
53 - I can add an article to the site
54 - I can set begin distributed dates, old article date and quit distributed dates for articles
55 - I need to have the capacity to assign regardless of whether an article ever makes the landing page

56 - I have entirely great control over how the article looks
57 - I need the connection from the article teaser to take me straightforwardly to the body of the article
58 - I need to have the capacity to demonstrate whether an article is freely accessible or for individuals as it were
59 - I need to have the capacity to peruse some of your articles
60 - I need to have full access to all articles
61 - I can do a full-message inquiry of article body, title, and writer name
62 - I can subscribe to a RSS channel of articles
63 - I can post remarks about articles so others can read them
64 - I need to have a conspicuous territory on the landing page where I can put unique declarations, not as a matter of course news or articles
65 - I might want to have some adaptability as to where things seem to oblige diverse sorts of substance
66 - I need advise the guests about up and coming courses
67 - I need to see new substance when I go to the site
68 - I need to have articles that intrigue me and are anything but difficult to get to
69 - I have thoughts on how I need the landing page to look and feel
70 - I have to know when I visit what on earth Scrum is, and why it needs a union
71 - I need to know as I look around the landing page what on earth a CSM is
72 - I need to have the capacity to return to the landing page rapidly and effectively
73 - I need to see a rundown of the most prevalent things on the site
74 - I am messaged a connection to a study about the course and teacher
75 - I need to be guaranteed that nobody can present the same answers numerous time and skew my outcomes
76 - I am informed about the aftereffects of studies about my classes
77 - I can see the outcomes for every coach and midpoints for the class
78 - I need to see a coach's evaluating
79 - I need my rating to appear on my profile page
80 - I need there to be a segment of the site that shows me the essentials of what Scrum is
81 - I can make the substance of the What Is Scrum segment
82 - I can see records on the site of all Certified Scrum Masters, Practitioners, Trainers, and Certified *Product* Owners
83 - I can have my name recorded in the registry without turning into an individual from the site
84 - I can stack an Excel record into the site
85 - I can see all classes in a pending state
86 - I can move individuals in his or her class from a pending state to the registry
87 - I am sent an email inviting me to the Scrum Alliance and containing directions on the best way to enlist/actuate my enrollment
88 - I can alter the substance of the email naturally sent to new Certified Scrum Masters and *Product* Owners
89 - I can join the Scrum Alliance by paying a corporate enrollment charge

90 - I need my logo is shown on a Corporate Sponsors page
91 - I need my logo to haphazardly show up on the landing page
92 - I am charged an expense
93 - I can pay a yearly charge
94 - I can set the yearly charges for individuals, Practitioners and Trainers
95 - I am sent an update and a connection through which I can restore
96 - I can have the framework email me another secret word or a watchword update and child on
97 - I can read data of pertinence just to coaches
98 - I can post data in a coaches just segment

Funcionalidades duplicadas geradas pela ferramenta Article Rewriter Tool para a base de estórias SAWeb (funcionalidades sublinhadas foram as escolhidas aleatoriamente para o experimento)

0 - I need to portray myself all alone page in a semi-organized manner
1 - I can round out an application to wind up a Practitioner
2 - I need my profile page to incorporate extra insights about me
3 - I can round out an application to wind up a Trainer
4 - I need my profile page to incorporate extra insights about me
5 - I need a little realistic connected with the substance showing I am a Practitioner or Trainer
6 - I need my profile to list my up and coming classes and incorporate a connection to a point by point page about each
7 - I can see the profiles of different individuals
8 - I can hunt down profiles taking into account a couple fields
9 - I can check my profile as private in which case just my name will show up
10 - I can check my email address as private regardless of the fact that whatever remains of my profile is definitely not
11 - I can send an email to any part through a structure
12 - I can read honing and preparing applications and support or reject them
13 - I can alter any site part profile
14 - I can read current news on the landing page
15 - I can get to old news that is no more on the landing page
16 - I can email news things to the supervisor
17 - I can set the accompanying dates on a news thing: Start Publishing Date, Old News Date, Stop Publishing Date

18 - I can subscribe to a RSS channel of news
19 - I can appoint need numbers to news things
20 - I can see a rundown of all up and coming Certification Courses
21 - I can see a rundown of all up and coming Other Courses
22 - I can see a rundown of every single up and coming Event
23 - I can make another course or occasion
24 - I am charged a posting expense for that action
25 - I can make an Other Course or Event that is not charged a posting expense so that the Scrum Alliance doesn't charge itself for Scrum Gatherings that it puts on
26 - I can set the posting charge per Other Course or Event
27 - I can redesign one of my current courses or occasions
28 - I can erase one of my courses or occasions
29 - I can duplicate one of my courses or occasions with the goal that I can make another one
30 - I can erase any course or occasion
31 - I can upgrade any course or occasion
32 - I can transform a course into an occasion or an occasion into a course
33 - I have a propelled seek alternative that gives me a chance to fill in a type of pursuit criteria
34 - I can tap on the name of mentor and be taken to the profile of coach
35 - I can subscribe to a RSS channel of up and coming courses and occasions
36 - I can read FAQs
37 - I can keep up a FAQ segment
38 - I can do a full-message inquiry of the FAQs
39 - I can download the most recent preparing material and system PDFs
40 - I can download presentations, PDFs, and so on Scrum that I can utilize
41 - I can look through a posting of employments
42 - I can post a help needed advertisement
43 - I have to support every help needed promotion before it gets to the site
44 - I am messaged at whatever point a vocation is submitted
45 - I can subscribe to a RSS channel of occupations accessible
46 - I can alter and erase help needed promotions
47 - I need occupations to quit distributed on the site 30 days in the wake of being posted
48 - I need to have the capacity to amplify a promotion for an additional 30 days by going by the site and overhauling the posting
49 - I need to be messaged an update with the goal that I can go amplify the promotion
50 - I need to peruse another article on the front page about once per week
51 - I can incorporate a teaser with every article

52 - I need to peruse the whole article
53 - I can add an article to the site
54 - I can set begin distributed dates, old article date and quit distributed dates for articles
55 - I need to have the capacity to assign regardless of whether an article ever makes the landing page
56 - I have truly great control over how the article looks
57 - I need the connection from the article teaser to take me straightforwardly to the body of the article
58 - I need to have the capacity to demonstrate whether an article is openly accessible or for individuals as it were
59 - I need to have the capacity to peruse some of your articles
60 - I need to have full access to all articles
61 - I can do a full-message pursuit of article body, title, and writer name
62 - I can subscribe to a RSS channel of articles
63 - I can post remarks about articles with the goal that others can read them
64 - I need to have a noticeable range on the landing page where I can put unique declarations, not as a matter of course news or articles
65 - I might want to have some adaptability as to where things seem to suit diverse sorts of substance
66 - I need advise the guests about forthcoming courses
67 - I need to see new substance when I go to the site
68 - I need to have articles that intrigue me and are anything but difficult to get to
69 - I have thoughts on how I need the landing page to look and feel
70 - I have to know when I visit what on earth Scrum is, and why it needs a collusion
71 - I need to know as I look around the landing page what on earth a CSM is
72 - I need to have the capacity to return to the landing page rapidly and effortlessly
73 - I need to see a rundown of the most well known things on the site
74 - I am messaged a connection to a study about the course and teacher
75 - I need to be guaranteed that nobody can present the same answers various time and skew my outcomes
76 - I am told about the aftereffects of overviews about my classes
77 - I can see the outcomes for every mentor and midpoints for the class
78 - I need to see a mentor's evaluating
79 - I need my rating to appear on my profile page
80 - I need there to be an area of the site that shows me the rudiments of what Scrum is
81 - I can make the substance of the What Is Scrum area
82 - I can see records on the site of all Certified Scrum Masters, Practitioners, Trainers, and Certified *Product* Owners
83 - I can have my name recorded in the registry without turning into an individual from the site
84 - I can stack an Excel record into the site
85 - I can see all classes in a pending state

86 - I can move individuals in his or her class from a pending state to the registry
87 - I am sent an email inviting me to the Scrum Alliance and containing directions on the best way to enroll/enact my participation
88 - I can alter the substance of the email consequently sent to new Certified Scrum Masters and *Product* Owners
89 - I can join the Scrum Alliance by paying a corporate participation expense
90 - I need my logo is shown on a Corporate Sponsors page
91 - I need my logo to arbitrarily show up on the landing page
92 - I am charged an expense
93 - I can pay a yearly expense
94 - I can set the yearly expenses for individuals, Practitioners and Trainers
95 - I am sent an update and a connection through which I can reestablish
96 - I can have the framework email me another watchword or a secret word update and child on
97 - I can read data of significance just to mentors
98 - I can post data in a mentors just area

Funcionalidades duplicadas geradas pela ferramenta Paraphrasing-Tool para a base de estórias RepBath (funcionalidades sublinhadas foram as escolhidas aleatoriamente para o experimento)

0 - I need to Deposit and keep up datasets through a basic web interface
1 - I need to Have a client interface that is well known to me
2 - I need to Deposit and keep up datasets through Pure
3 - I need to Deposit and keep up datasets through Virtual Research Environments and other work process devices
4 - I need to Deposit the documents that I have
5 - I need to Place information under a ban
6 - I need to Apply licenses to datasets
7 - I need to Allow my colleagues special access to datasets
8 - I need to Deposit discretionarily vast documents
9 - I need to Link datasets to distributions in Opus
10 - I need to Mint DOIs for my information
11 - I need to Have metadata naturally filled from other University frameworks (e.g. Pure) and/or recalled from past stores
12 - I need to Link to information put away in outer stores
13 - I need to Specify a maintenance/transfer strategy for my information
14 - I need to Track downloads of my information

15 - I need to Track references of my information
16 - I need to Have ensures about information respectability
17 - I need to Attach subject-particular discoverability metadata to records
18 - I need to Link datasets with the venture DMP (conceivably from DMP online)
19 - I need to Manage and share "live" research information
20 - I need to Manage numerous adaptations of the same dataset
21 - I need to Allow others to store for my benefit
22 - I need to Search the document through the web
23 - I need to Access the framework in my local dialect
24 - I need to Examine and distinguish kept documents
25 - I need to View an illustration reference for a dataset
26 - I need to View a DOI for a dataset
27 - I need to Get a tenacious URL for a dataset
28 - I need to Search the document through Primo (University of Bath library look framework)
29 - I need to See distinctive variants (counting the most recent) of a dataset initially
30 - I need to Gain special access to information for tasks in which I am included
31 - I need to Have ensures that my IP rights won't be ruptured
32 - I need to Access information from Bath associates off grounds
33 - I need to Deposit information from my office straightforwardly into the chronicle for the benefit of analysts
34 - I need to Make a few keeps an eye on saved datasets before they are made open
35 - I need to Require a base arrangement of metadata
36 - I need to Approve booked transfer of information
37 - I need to Query the whole document (counting restricted records)
38 - I need to Import Bath information from an outer server farm wholesale
39 - I need to Encourage and advance the utilization of open gauges for store
40 - I need to Integrate the document with CRIS
41 - I need to Include records for remotely held information
42 - I need to Track reference means distributed datasets
43 - I need to Segment view and download insights by nation and area
44 - I need to Have datasets connected to metadata about activities
45 - I need to Store filed information on existing stockpiling frameworks
46 - I need to Integrate the file with existing college frameworks, for example, LDAP
47 - I need to Store filed information straightforwardly on the HCP article store
48 - I need to Be ready to send out all information to an alternate framework

49 - I need to Deposit and keep up datasets through an API, for example, SWORD2
50 - I need to Make relentless web connections between my articles and fundamental datasets
51 - I need to Be consoled (by individual specialists or an organization) that analysts I store have hearty archival arrangements for their information
52 - I need to Harvest metadata on yields from examination I support through e.g. OAI-PMH

Funcionalidades duplicadas geradas pela ferramenta Free Article Spinner para a base de estórias RepBath (funcionalidades sublinhadas foram as escolhidas aleatoriamente para o experimento)

0 - I need to Deposit and keep up datasets through a straightforward web interface
1 - I need to Have a client interface that is well known to me
2 - I need to Deposit and keep up datasets through Pure
3 - I need to Deposit and keep up datasets through Virtual Research Environments and other work process apparatuses
4 - I need to Deposit the documents that I have
5 - I need to Place information under a ban
6 - I need to Apply licenses to datasets
7 - I need to Allow my partners advantaged access to datasets
8 - I need to Deposit self-assertively substantial documents
9 - I need to Link datasets to distributions in Opus
10 - I need to Mint DOIs for my information
11 - I need to Have metadata naturally filled from other University frameworks (e.g. Unadulterated) and/or recalled from past stores
12 - I need to Link to information put away in outside vaults
13 - I need to Specify a maintenance/transfer approach for my information
14 - I need to Track downloads of my information
15 - I need to Track references of my information
16 - I need to Have ensures about information honesty
17 - I need to Attach subject-particular discoverability metadata to records
18 - I need to Link datasets with the undertaking DMP (perhaps from DMP online)
19 - I need to Manage and share "live" research information
20 - I need to Manage numerous renditions of the same dataset
21 - I need to Allow others to store for my benefit
22 - I need to Search the document through the web

23 - I need to Access the framework in my local dialect
24 - I need to Examine and distinguish kept records
25 - I need to View an illustration reference for a dataset
26 - I need to View a DOI for a dataset
27 - I need to Get a tenacious URL for a dataset
28 - I need to Search the document through Primo (University of Bath library look framework)
29 - I need to See diverse renditions (counting the most recent) of a dataset initially
30 - I need to Gain special access to information for activities in which I am included
31 - I need to Have ensures that my IP rights won't be broken
32 - I need to Access information from Bath associates off grounds
33 - I need to Deposit information from my office straightforwardly into the file for specialists
34 - I need to Make a few minds kept datasets before they are made open
35 - I need to Require a base arrangement of metadata
36 - I need to Approve booked transfer of information
37 - I need to Query the whole chronicle (counting restricted records)
38 - I need to Import Bath information from an outer server farm wholesale
39 - I need to Encourage and advance the utilization of open models for store
40 - I need to Integrate the chronicle with CRIS
41 - I need to Include records for remotely held information
42 - I need to Track reference means distributed datasets
43 - I need to Segment view and download insights by nation and division
44 - I need to Have datasets connected to metadata about ventures
45 - I need to Store chronicled information on existing stockpiling frameworks
46 - I need to Integrate the chronicle with existing college frameworks, for example, LDAP
47 - I need to Store chronicled information straightforwardly on the HCP object store
48 - I need to Be ready to fare all information to an alternate framework
49 - I need to Deposit and keep up datasets through an API, for example, SWORD2
50 - I need to Make industrious web joins between my articles and fundamental datasets
51 - I need to Be consoled (by individual analysts or an organization) that specialists I finance have powerful recorded arrangements for their information
52 - I need to Harvest metadata on yields from exploration I support by means of e.g. OAI-PMH

Funcionalidades duplicadas geradas pela ferramenta Article Rewriter Tool para a base de estórias RepBath (funcionalidades sublinhadas foram as escolhidas aleatoriamente para o experimento)

0 - I need to Deposit and keep up datasets through a straightforward web interface
1 - I need to Have a client interface that is well known to me
2 - I need to Deposit and keep up datasets through Pure
3 - I need to Deposit and keep up datasets through Virtual Research Environments and other work process devices
4 - I need to Deposit the documents that I have
5 - I need to Place information under a ban
6 - I need to Apply licenses to datasets
7 - I need to Allow my partners special access to datasets
8 - I need to Deposit self-assertively expansive records
9 - I need to Link datasets to distributions in Opus
10 - I need to Mint DOIs for my information
11 - I need to Have metadata naturally filled from other University frameworks (e.g. Immaculate) and/or recollected from past stores
12 - I need to Link to information put away in outside vaults
13 - I need to Specify a maintenance/transfer arrangement for my information
14 - I need to Track downloads of my information
15 - I need to Track references of my information
16 - I need to Have ensures about information trustworthiness
17 - I need to Attach subject-particular discoverability metadata to records
18 - I need to Link datasets with the venture DMP (perhaps from DMP online)
19 - I need to Manage and share "live" research information
20 - I need to Manage numerous forms of the same dataset
21 - I need to Allow others to store for my benefit
22 - I need to Search the document through the web
23 - I need to Access the framework in my local dialect
24 - I need to Examine and distinguish stored records
25 - I need to View an illustration reference for a dataset
26 - I need to View a DOI for a dataset
27 - I need to Get an industrious URL for a dataset
28 - I need to Search the document through Primo (University of Bath library seek framework)

29 - I need to See diverse variants (counting the most recent) of a dataset initially
30 - I need to Gain special access to information for ventures in which I am included
31 - I need to Have ensures that my IP rights won't be broken
32 - I need to Access information from Bath teammates off grounds
33 - I need to Deposit information from my office straightforwardly into the file in the interest of specialists
34 - I need to Make a few keeps an eye on stored datasets before they are made open
35 - I need to Require a base arrangement of metadata
36 - I need to Approve booked transfer of information
37 - I need to Query the whole chronicle (counting banned records)
38 - I need to Import Bath information from an outer server farm wholesale
39 - I need to Encourage and advance the utilization of open models for store
40 - I need to Integrate the chronicle with CRIS
41 - I need to Include records for remotely held information
42 - I need to Track reference means distributed datasets
43 - I need to Segment view and download measurements by nation and area
44 - I need to Have datasets connected to metadata about activities
45 - I need to Store filed information on existing stockpiling frameworks
46 - I need to Integrate the file with existing college frameworks, for example, LDAP
47 - I need to Store filed information straightforwardly on the HCP object store
48 - I need to Be ready to fare all information to an alternate framework
49 - I need to Deposit and keep up datasets by means of an API, for example, SWORD2
50 - I need to Make diligent web joins between my articles and hidden datasets
51 - I need to Be consoled (by individual scientists or an establishment) that analysts I finance have vigorous chronicled plans for their information
52 - I need to Harvest metadata on yields from examination I finance through e.g. OAI-PMH

Funcionalidades duplicadas geradas pela ferramenta Paraphrasing-Tool para a base de estórias COTUCA (funcionalidades sublinhadas foram as escolhidas aleatoriamente para o experimento)

0 - I need to send messages to pupils
1 - I need to send the report card of pupils by email to folks
2 - I need to send messages to pupils
3 - I need to send documents to permit pupils to download
4 - I need to send documents to permit educators to download
5 - I need to send messages to organizations
6 - I need to send messages to pupils
7 - I need to take after the preparation contracts
8 - I need to sign new preparing contracts
9 - I need to question the information of organizations and consented to arrangements for choice making
10 - I need to roll out improvements in preparing contracts
11 - I need to fare preparing contracts in PDF arrangement to get to them anyplace
12 - I need to get the temporary job reports in PDF configuration to effectively circulate them to educators
13 - I need to send declarations and news for pupils
14 - I need to produce watchword to distribute to pupils
15 - I need to produce watchword to distribute to folks
16 - I need to see pupils who conveyed stages reports to draft conclusion confirmations
17 - I need to send declarations to the instructors
18 - I need to counsel my notes
19 - I need to get notification and news
20 - I need to enlisting in courses
21 - I need to enlisting in entry level position discipline
22 - I need to send my temporary position report
23 - I need to change my secret word
24 - I need to see my transcript
25 - I need to counsel the subjects that still need to go to
26 - I need to print my transcript
27 - I need to print enlistment testament
28 - I need to wipe out my enlistment in any order
29 - I need to see the outcome application for enrollment for courses

30 - I need to see my GPA;
31 - I need to counsel my entry level position contract
32 - I need to sign my entry level position contract electronically
33 - I need to change my enrollment information
34 - I need to check the timetables of the subjects of my course and different courses
35 - I need to print an enrollment accomplishment of verification
36 - I need to counsel the course educational modules that I am selected for the courses and essentials
37 - I need to downloading of basic utilization of archives
38 - I need to downloading of pedantic material of the course
39 - I need to trade my transcript in PDF group
40 - I need to send out my enrollment testament in PDF design
41 - I need to ask for a brief moment duplicate of understudy card
42 - I need to demand archives and administrations to the workplace
43 - I need to get notices about entry level position opportunities
44 - I need illuminate pupils' evaluations
45 - I need counsel some information from my pupils
46 - I need record the school day by day trains that educate
47 - I need check pupils' evaluations in different subjects
48 - I need change my watchword
49 - I need fare evaluations of pupils in CSV or XLS group
50 - I need send email to pupils
51 - I need send email to the understudy's guardians
52 - I need give instructive material to pupils
53 - I need print the school journal of my subjects
54 - I need send out electronic diaries in PDF
55 - I need downloading of photos of pupils
56 - I need convey my school day by day on the web
57 - I need download the temporary job reports
58 - I need convey last grades by means of the Internet
59 - I need halfway conveyance notes over the Internet
60 - I need download the lesson arranges in PDF group
61 - I need enrolling singular appraisal of understudy data
62 - I need counsel the recurrence of pupils in medicinal classes
63 - I need counsel insights of the notes in a class

64 - I need download regularly utilized records of instructors
65 - I need reserve classroom spot
66 - I need send declarations to the pupils
67 - I need send declarations to the pupils
68 - I need get the report card by email
69 - I need see messages or notices given by the instructor to me
70 - I need empower or handicap the receipt of interchanges by e-mail

Funcionalidades duplicadas geradas pela ferramenta Free Article Spinner para a base de estórias COTUCA (funcionalidades sublinhadas foram as escolhidas aleatoriamente para o experimento)

0 - I need to send messages to understudies
1 - I need to send the report card of understudies by email to folks
2 - I need to send messages to understudies
3 - I need to send records to permit understudies to download
4 - I need to send records to permit instructors to download
5 - I need to send messages to organizations
6 - I need to send messages to understudies
7 - I need to take after the preparation contracts
8 - I need to sign new preparing contracts
9 - I need to question the information of organizations and consented to arrangements for basic leadership
10 - I need to roll out improvements in preparing contracts
11 - I need to fare preparing contracts in PDF arrangement to get to them anyplace
12 - I need to get the temporary position reports in PDF arrangement to effortlessly disseminate them to instructors
13 - I need to send declarations and news for understudies
14 - I need to create secret key to give out to understudies
15 - I need to create secret key to give out to folks
16 - I need to see understudies who conveyed stages reports to draft conclusion recognitions
17 - I need to send declarations to the educators
18 - I need to counsel my notes
19 - I need to get notification and news

20 - I need to selecting in courses
21 - I need to selecting in entry level position discipline
22 - I need to send my temporary position report
23 - I need to change my secret word
24 - I need to see my transcript
25 - I need to counsel the subjects that even now need to go to
26 - I need to print my transcript
27 - I need to print enrollment declaration
28 - I need to scratch off my enlistment in any order
29 - I need to see the outcome application for enrollment for courses
30 - I need to see my GPA
31 - I need to counsel my temporary job contract
32 - I need to sign my temporary job contract electronically
33 - I need to change my enrollment information
34 - I need to check the timetables of the subjects of my course and different courses
35 - I need to print an enrollment accomplishment of verification
36 - I need to counsel the course educational modules that I am selected for the courses and requirements
37 - I need to downloading of basic utilization of archives
38 - I need to downloading of pedantic material of the course
39 - I need to send out my transcript in PDF group
40 - I need to send out my enrollment declaration in PDF position
41 - I need to ask for a moment duplicate of understudy card
42 - I need to demand reports and administrations to the workplace
43 - I need to get notices about temporary job opportunities
44 - I need advise understudies' evaluations
45 - I need counsel a few information from my understudies
46 - I need record the school every day trains that educate
47 - I need check understudies' evaluations in different subjects
48 - I need change my secret word
49 - I need send out evaluations of understudies in CSV or XLS group
50 - I need send email to understudies
51 - I need send email to the understudy's guardians
52 - I need give instructive material to understudies
53 - I need print the school journal of my subjects

54 - I need send out electronic diaries in PDF
55 - I need downloading of photos of understudies
56 - I need convey my school every day on the web
57 - I need download the temporary position reports
58 - I need convey last grades through the Internet
59 - I need fractional conveyance notes over the Internet
60 - I need download the lesson arranges in PDF position
61 - I need enrolling singular appraisal of understudy data
62 - I need counsel the recurrence of understudies in therapeutic classes
63 - I need counsel insights of the notes in a class
64 - I need download generally utilized documents of instructors
65 - I need reserve classroom spot
66 - I need send declarations to the understudies
67 - I need send declarations to the understudies
68 - I need get the report card by email
69 - I need see messages or notices given by the educator to me
70 - I need empower or cripple the receipt of correspondences by e-mail

Funcionalidades duplicadas geradas pela ferramenta Article Rewriter Tool para a base de estórias COTUCA (funcionalidades sublinhadas foram as escolhidas aleatoriamente para o experimento)

0 - I need to send messages to understudies
1 - I need to send the report card of understudies by email to folks
2 - I need to send messages to understudies
3 - I need to send records to permit understudies to download
4 - I need to send records to permit educators to download
5 - I need to send messages to organizations
6 - I need to send messages to understudies
7 - I need to take after the preparation contracts
8 - I need to sign new preparing contracts
9 - I need to question the information of organizations and consented to arrangements for basic leadership

10 - I need to roll out improvements in preparing contracts
11 - I need to fare preparing contracts in PDF arrangement to get to them any place
12 - I need to get the temporary job reports in PDF organization to effectively circulate them to educators
13 - I need to send declarations and news for understudies
14 - I need to create secret word to give out to understudies
15 - I need to create secret word to give out to folks
16 - I need to see understudies who conveyed stages reports to draft conclusion certificates
17 - I need to send declarations to the educators
18 - I need to counsel my notes
19 - I need to get notification and news
20 - I need to selecting in courses
21 - I need to selecting in temporary job discipline
22 - I need to send my entry level position report
23 - I need to change my secret key
24 - I need to see my transcript
25 - I need to counsel the subjects that at present need to go to
26 - I need to print my transcript
27 - I need to print enrollment authentication
28 - I need to cross out my enlistment in any control
29 - I need to see the outcome application for enrollment for courses
30 - I need to see my GPA
31 - I need to counsel my entry level position contract
32 - I need to sign my entry level position contract electronically
33 - I need to change my enrollment information
34 - I need to check the timetables of the subjects of my course and different courses
35 - I need to print an enrollment accomplishment of confirmation
36 - I need to counsel the course educational modules that I am selected for the courses and requirements
37 - I need to downloading of regular utilization of reports
38 - I need to downloading of educational material of the course
39 - I need to send out my transcript in PDF design
40 - I need to trade my enlistment endorsement in PDF group
41 - I need to ask for a brief moment duplicate of understudy card
42 - I need to demand records and administrations to the workplace
43 - I need to get notices about temporary position opportunities

44 - I need advise understudies' evaluations
45 - I need counsel a few information from my understudies
46 - I need record the school every day trains that instruct
47 - I need check understudies' evaluations in different subjects
48 - I need change my secret word
49 - I need send out evaluations of understudies in CSV or XLS group
50 - I need send email to understudies
51 - I need send email to the understudy's guardians
52 - I need give instructive material to understudies
53 - I need print the school journal of my subjects
54 - I need send out electronic diaries in PDF
55 - I need downloading of photos of understudies
56 - I need convey my school day by day on the web
57 - I need download the temporary job reports
58 - I need convey last grades by means of the Internet
59 - I need halfway conveyance notes over the Internet
60 - I need download the lesson arranges in PDF group
61 - I need enlisting singular evaluation of understudy data
62 - I need counsel the recurrence of understudies in healing classes
63 - I need counsel insights of the notes in a class
64 - I need download normally utilized documents of instructors
65 - I need reserve classroom spot
66 - I need send declarations to the understudies
67 - I need send declarations to the understudies
68 - I need get the report card by email
69 - I need see messages or notices given by the educator to me
70 - I need empower or debilitate the receipt of correspondences by e-mail

Apêndice C

Funcionalidades geradas para detecção de evolução para as estórias de usuário da base SAWeb

1 - I want to include in my profile details about the courses that I attended
2 - I want to view the profiles of trainers the same way I see the profiles of members
3 - I want to search profiles using a language to query
4 - I want my name and email appears when my profile is private
5 - I want to receive a confirmation email when I send an email via form
6 - I can edit only Practitioner profile
7 - I want to edit the course or event added by me
8 - I want to include answer to each teaser
9 - I want to put a publication date to the articles added by me
10 - I want to receive a PDF file by e-mail with data of surveys about my classes
11- I want my rating to appear on general rating page
12 - I want to load Excel file and keep it visible only to me
13 - I want to see all classes of any state
14 - I can receive a confirmation payment of my annual fee
15 - I can read information submitted to me by students

Funcionalidades geradas para detecção de evolução para as estórias de usuário da base RepBath

1 - As a depositor I want to share the file that I deposited;
2 - As a depositor I want to see a list of downloads of my data;
3 - As a depositor I want to manage only the last ten versions of a dataset;
4 - As a data re-user I want to choose the language to access the system;
5 - As a data re-user I want to do backup of deposited files;
6 - As a Bath Data Archive administrator I want to recover data disposed by system of disposal of data;
7 - As a Research Information manager I want to see a list of the most cited dataset published;
8 - As the university IT service I want export data in PDF and XLS format;

Funcionalidades geradas para detecção de evolução para as estórias de usuário da base COTUCA

1 - I want to change the password to students
2 - I want to consult the courses and disciplines that I am enrolled
3 - I want to receive a confirmation by email when I cancel my enrollment in some discipline
4 - I want to export my transcript in DOC format
5 - I want to print the ticket payment of second via of my student card
6 - I want to share my daily school with other teachers
7 - I want that system to inform students when I submit new educational materials
8 - I want to export my school daily in XLS format
9 - I want to undo delivery my school daily on the internet
10 - I want to cancel the reservation of a classroom made by me
11 - I want to schedule the sending of announcements to the students

Apêndice D

Estórias de usuário da base SAWeb distribuída pelo analista humano

Nº	Business Value	Story Points	Sprint	User story
0	2 - Could have	8 - Grande	7	As a site member I want to describe myself on my own page in a semi-structured way
1	3- Should have	5 - Média	3	As a site member I can fill out an application to become a Practitioner
2	2 - Could have	8 - Grande	7	As a Practitioner I want my profile page to include additional details about me
3	3- Should have	5 - Média	3	As a site member I can fill out an application to become a Trainer
4	3- Should have	3 - Pequena	2	As a Trainer I want my profile page to include additional details about me
5	2 - Could have	2 - Muito pequena	5	As a Practitioner or Trainer I want a small graphic associated with the content indicating I am a Practitioner or Trainer
6	3- Should have	5 - Média	3	As a trainer I want my profile to list my upcoming classes and include a link to a detailed page about each
7	3- Should have	3 - Pequena	2	As a site member I can view the profiles of other members
8	2 - Could have	5 - Média	6	As a site member I can search for profiles based on a few fields
9	1 - Won't have	5 - Média	8	As a site member I can mark my profile as private in which case only my name will appear
10	1 - Won't have	3 - Pequena	7	As a site member I can mark my email address as private even if the rest of my profile is not
11	2 - Could have	5 - Média	6	As a site member I can send an email to any member via a form
12	3- Should have	8 - Grande	4	As a site administrator I can read practicing and training applications and approve or reject them
13	3- Should have	5 - Média	3	As a site administrator I can edit any site member profile
14	4 - Must have	2 - Muito pequena	1	As a site visitor I can read current news on the home page
15	3- Should have	3 - Pequena	2	As a site visitor I can access old news that is no longer on the home page
16	1 - Won't have	5 - Média	8	As a site visitor I can email news items to the editor
17	3- Should have	8 - Grande	4	As site editor I can set the following dates on a news item: Start Publishing Date, Old News Date, Stop Publishing Date
18	3- Should have	5 - Média	3	As a site member I can subscribe to an RSS feed of news
19	2 - Could have	3 - Pequena	5	As a site editor I can assign priority numbers to news items
20	4 - Must have	5 - Média	1	As a site visitor I can see a list of all upcoming Certification Courses

21	3- Should have	5 - Média	3	As a site visitor I can see a list of all upcoming Other Courses
22	3- Should have	5 - Média	3	As a site visitor I can see a list of all upcoming Events
23	3- Should have	8 - Grande	4	As a trainer I can create a new course or event
24	4 - Must have	5 - Média	1	As a trainer I am charged a listing fee for that activity
25	2 - Could have	8 - Grande	7	As a site administrator I can create an Other Course or Event that is not charged a listing fee so that the Scrum Alliance doesn't charge itself for Scrum Gatherings that it puts on
26	4 - Must have	5 - Média	1	As a site administrator I can set the listing fee per Other Course or Event
27	3- Should have	5 - Média	3	As a trainer I can update one of my existing courses or events
28	1 - Won't have	3 - Pequena	8	As a trainer I can delete one of my courses or events
29	2 - Could have	8 - Grande	7	As a trainer I can copy one of my courses or events so that I can create a new one
30	3- Should have	3 - Pequena	2	As a site admin I can delete any course or event
31	3- Should have	5 - Média	3	As a site editor I can update any course or event
32	2 - Could have	5 - Média	6	As a trainer, admin, or editor, I can turn a course into an event or an event into a course
33	1 - Won't have	8 - Grande	9	As a site visitor I have an advanced search option that lets me fill in a form of search criteria
34	3- Should have	3 - Pequena	2	As a site visitor I can click on the name of trainer and be taken to the profile of trainer
35	3- Should have	3 - Pequena	2	As a site visitor I can subscribe to an RSS feed of upcoming courses and events
36	2 - Could have	2 - Muito pequena	5	As a site visitor I can read FAQs
37	2 - Could have	3 - Pequena	6	As a site editor I can maintain an FAQ section
38	2 - Could have	5 - Média	6	As a site member I can do a full-text search of the FAQs
39	1 - Won't have	5 - Média	8	As a site member I can download the latest training material and methodology PDFs
40	1 - Won't have	5 - Média	8	As a visitor I can download presentations, PDFs, etc. on Scrum that I can use
41	3- Should have	8 - Grande	4	As a site member I can scroll through a listing of jobs
42	2 - Could have	8 - Grande	7	As someone who wants to hire I can post a help wanted ad
43	2 - Could have	5 - Média	6	As a site admin I need to approve each help wanted ad before it gets to the site
44	2 - Could have	3 - Pequena	6	As a site admin I am emailed whenever a job is submitted
45	2 - Could have	3 - Pequena	6	As a site member I can subscribe to an RSS feed of jobs available
46	3- Should have	3 - Pequena	2	As a site admin I can edit and delete help wanted ads
47	3- Should have	3 - Pequena	2	As a site admin I want jobs to stop publishing on the site 30 days after being posted
48	3- Should have	8 - Grande	3	As someone who wants to hire I want to be able to extend an ad for another 30 days by visiting the site and

				updating the posting
49	3- Should have	3 - Pequena	2	As someone who has posted an ad that is about to expire, seven days before it expires I want to be emailed a reminder so that I can go extend the ad
50	2 - Could have	5 - Média	6	As a site visitor I want to read a new article on the front page about once a week
51	3- Should have	5 - Média	4	As the site editor I can include a teaser with each article
52	3- Should have	3 - Pequena	2	As a site member who has read a teaser on the front page I want to read the entire article
53	4 - Must have	5 - Média	1	As the site editor I can add an article to the site
54	2 - Could have	8 - Grande	8	As a site editor I can set start publishing dates, old article date and stop publishing dates for articles
55	3- Should have	3 - Pequena	3	As a site editor I want to be able to designate whether or not an article ever makes the home page
56	3- Should have	40 - Enorme	5	As the site editor I have pretty good control over how the article looks
57	3- Should have	3 - Pequena	2	As a site visitor I want the link from the article teaser to take me directly to the body of the article
58	3- Should have	5 - Média	2	As a site editor I want to be able to indicate whether an article is publicly available or for members only
59	4 - Must have	1 - Minúscula	1	As a site visitor I want to be able to read some of your articles
60	3- Should have	5 - Média	2	As a site member I want to have full access to all articles
61	1 - Won't have	5 - Média	9	As a site visitor I can do a full-text search of article body, title, and author name
62	2 - Could have	3 - Pequena	6	As a site visitor I can subscribe to an RSS feed of articles
63	1 - Won't have	8 - Grande	9	As a site visitor I can post comments about articles so that others can read them
64	2 - Could have	5 - Média	6	As a site editor I want to have a prominent area on the home page where I can put special announcements, not necessarily news or articles
65	2 - Could have	13 - Muito grande	8	As a site editor I would like to have some flexibility as to where things appear to accommodate different types of content
66	3- Should have	2 - Muito pequena	2	As a site member I want notify the visitors about upcoming courses
67	3- Should have	1 - Minúscula	2	As a site visitor I want to see new content when I come to the site
68	3- Should have	2 - Muito pequena	2	As a site visitor I want to have articles that interest me and are easy to get to
69	2 - Could have	13 - Muito grande	8	As a site editor I have ideas on how I want the home page to look and feel
70	4 - Must have	2 - Muito pequena	1	As a site visitor I need to know as soon as I visit what on earth Scrum is, and why it needs an alliance
71	4 - Must have	2 - Muito pequena	1	As a site visitor I want to know as I glance around the home page what on earth a CSM is and why I would want to be one

72	2 - Could have	1 - Minúscula	5	As a site visitor I want to be able to get back to the home page quickly and easily
73	2 - Could have	5 - Média	6	As a site visitor I want to see a list of the most popular items on the site
74	3- Should have	3 - Pequena	2	As someone who successfully completed a Certification Course I am emailed a link to a survey about the course and instructor
75	2 - Could have	5 - Média	6	As a trainer I want to be assured that no one can submit the same answers multiple time and skew my results
76	3- Should have	3 - Pequena	2	As a trainer I am notified about the results of surveys about my classes
77	3- Should have	5 - Média	3	As a site admin I can see the results for each trainer and averages for the class
78	2 - Could have	3 - Pequena	6	As a site visitor who is considering attending a certification course I want to see a trainer's rating
79	3- Should have	5 - Média	3	As a trainer I want my rating to show up on my profile page
80	2 - Could have	8 - Grande	7	As a site visitor I want there to be a section of the website that teaches me the basics of what Scrum is
81	3- Should have	5 - Média	4	As a site editor I can create the content of the What Is Scrum section
82	2 - Could have	5 - Média	7	As a site visitor I can view lists on the site of all Certified ScrumMasters, Practitioners, Trainers, and Certified Product Owners
83	2 - Could have	2 - Muito pequena	5	As a CSM, Practitioner or Certified Product Owner I can have my name listed in the registry without becoming a member of the site
84	2 - Could have	5 - Média	7	As a trainer who has finished teaching a Certification class I can load an Excel file into the site
85	3- Should have	3 - Pequena	2	As a site admin I can view all classes in a pending state
86	3- Should have	5 - Média	4	As a site admin who has received proof of payment from a trainer I can move people in his or her class from a pending state to the registry
87	4 - Must have	3 - Pequena	1	As a new Certified ScrumMaster or Certified Product Owner I am sent an email welcoming me to the Scrum Alliance and containing instructions on how to register / activate my membership
88	3- Should have	5 - Média	4	As a site editor I can edit the content of the email automatically sent to new Certified ScrumMasters and Product Owners
89	4 - Must have	8 - Grande	1	As a company I can join the Scrum Alliance by paying a corporate membership fee
90	4 - Must have	1 - Minúscula	1	As a corporate sponsor I want my logo is displayed on a Corporate Sponsors page
91	4 - Must have	2 - Muito pequena	1	As a corporate sponsor I want my logo to randomly appear on the home page
92	4 - Must have	8 - Grande	1	As a Certified ScrumMaster or Certified Product Owner who has been approved for Practitioner status I am charged a fee
93	4 - Must have	8 - Grande	1	As someone about to become a trainer I can pay an annual fee
94	3- Should have	5 - Média	4	As a site administrator I can set the annual fees for members, Practitioners and Trainers
95	3- Should have	8 - Grande	5	As someone whose membership is about to expire I am sent a reminder and a link through which I can renew

96	4 - Must have	5 - Média	1	As a member with short-term memory problems I can have the system email me a new password or a password reminder and son on
97	2 - Could have	3 - Pequena	6	As a trainer I can read information of relevance only to trainers
98	3- Should have	5 - Média	4	As a site editor I can post information in a trainers-only section

Índice remissivo