

Criatividade Combinacional para Geração de Requisitos Inovadores: Um Relato de Experiência

Rafael Pinto^{1,2}, Lyrene Silva¹, Marcia Lucena¹ e Ilueny Santos¹

¹ Universidade Federal do Rio Grande do Norte,
Centro de Ciências Exatas,

Departamento de Informática e Matemática Aplicada.
BR 101 - Lagoa Nova, 59078970 - Natal/RN - Brasil

² Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte,
Reitoria, Coordenação de Sistemas de Informação
Rua Dr. Nilo Bezerra Ramalho, 1692, Tirol, 59015300 - Natal/RN - Brasil
{rrafaelpinto, lyrene, marcia.lucena, ilueny}@gmail.com

Resumo A Engenharia de Requisitos tem buscado, através de técnicas de criatividade, fornecer meios para elicitação de requisitos úteis e inovadores. Dentre as técnicas disponíveis, há a de Criatividade Combinacional, que visa produzir novas ideias através da associação de ideias antigas utilizando caminhos não familiares. O presente trabalho tem por objetivo propor uma estratégia baseada na técnica de criatividade combinacional para criar novos requisitos, utilizando como fonte de dados a documentação do próprio sistema. Um experimento, com uma documentação de um sistema real, foi realizado por alunos do Bacharelado em Engenharia de Software da Universidade Federal do Rio Grande do Norte (UFRN). Ao analisar as respostas, constatamos que 44% das ideias propostas pelos participantes podem ser classificadas como originais e relevantes para o sistema. Dessa forma, os resultados mostram que nossa estratégia pode enriquecer abordagens existentes de forma a complementar a elicitação de requisitos.

Keywords: Engenharia de Requisitos, Elicitação de Requisitos, Criatividade Combinacional

1 Introdução

A qualidade da definição dos requisitos é considerada um dos fatores de sucesso no desenvolvimento de sistemas de software, visto que os requisitos são a base para todo o processo de desenvolvimento [1]. A elicitação é considerada uma das tarefas mais difíceis da engenharia de requisitos, já que envolve a descoberta do que o usuário realmente necessita [2]. Isso porque, em muitos casos, o próprio cliente não sabe ou não consegue expressar tudo o que é necessário e esperado do sistema a ser desenvolvido. Dessa forma, é de fundamental importância usar as técnicas de elicitação mais adequadas ao contexto e fontes de informações disponíveis. Com isso, a Engenharia de Requisitos tem buscado e usado técnicas

de criatividade como meio complementar para elicitaco de requisitos teis e relevantes.

Segundo Sternberg [3], a criatividade  definida como “a competncia de produzir um trabalho que  ao mesmo tempo inovador (ou seja, original e inesperado) e apropriado (isto , til e adaptvel para as atividades requeridas)”. A capacidade de produzir novas ideias atravs da associao de ideias antigas  denominada criatividade combinacional. Este tipo de criatividade requer no s uma rica fonte de ideias antigas, mas a habilidade de fazer associao de muitos tipos diferentes [4].

Por outro lado, para Pennel e Maiden [5], no contexto de requisitos,  mais importante habilitar os participantes a produzir ideias para requisitos que normalmente no seriam elicitados do que gerar requisitos tidos como inovadores.

Nosso trabalho foi inspirado no artigo de Bhowmik e outros [6], que prope a criao de um *framework* que extrai ideias familiares dos requisitos e comentrios dos *stakeholders* registrados em uma rede social ou sistema de acompanhamento de tarefas (Bugzilla, por exemplo). Para realizar a associao de tipos diferentes de ideias, Bhowmik e outros utilizam algoritmos de Processamento de Linguagem Natural (PLN) para descobrir a estrutura temtica (tpicos) por trs de grandes textos. Aps a descoberta destes tpicos, pares de palavras (substantivos e verbos) so separados e disponibilizados aos especialistas no domnio da aplicao, afim de que possam propor requisitos inovadores.

Nossa proposta se diferencia da abordagem original em alguns aspectos: (i) A fonte de dados  puramente a base de requisitos (documentao) do sistema em questo. Com isso o agrupamento das ideias familiares so extradas de um mdulo especfico do sistema; (ii) a lista de palavras processadas so dispostas em formato de nuvem, o que implica em uma disposio diferenciada das palavras alm de permitir ao *stakeholder* escolher os pares de substantivo e verbo livremente; (iii) este trabalho foi realizado visando o processamento de palavras em lngua portuguesa.

O presente trabalho tem por objetivo apresentar uma estratgia para criar novos e relevantes requisitos atravs da combinao no familiar de palavras, utilizando como fonte de dados a documentao textual de requisitos e funcionalidades do prprio sistema. Alm disso, relatamos e discutimos os resultados alcanados com um estudo de caso realizado com usurios de um sistema de controle acadmico da Universidade Federal do Rio Grande do Norte (UFRN).

Aps um breve resumo sobre a criatividade na engenharia de requisitos, bem como o tipo de criatividade abordado por este trabalho (seo 2), iremos apresentar o processo usado em nossa estratgia (seo 3). Na seo 4, relatamos como aplicamos nossa estratgia em um estudo de caso real. Na ltima seo (5) mostramos as considerao finais do trabalho e os possveis trabalhos futuros a serem desenvolvidos.

2 Fundamentação Teórica

2.1 Criatividade na Engenharia de Requisitos

Atualmente, a indústria de software tem sido estimulada a desenvolver inovações para se diferenciar dos concorrentes e atrair novos clientes [7]. No passado, os requisitos de software eram considerados implícitos na mente dos *stakeholders* e o principal trabalho do analista era de conhecer suas necessidades e desejos usando as várias técnicas de elicitação. Mas para Maiden [8] e Robertson [9] este ponto de vista não é mais verdade. Para eles, a criatividade em Engenharia de Requisitos (ER) oferece grande vantagem competitiva para os produtos.

Na última década várias pesquisas relacionadas à criatividade na ER foram desenvolvidas. Como podemos observar em [7], um estudo apresentado com o objetivo de realizar um mapeamento de toda literatura desenvolvida neste campo fez a análise de 46 trabalhos. Os resultados destacaram os 4 principais grupos de pesquisa que atuam em criatividade na ER. São eles:

- **N. Maiden/S. Robertson e outros:** a principal contribuição deste grupo é desenvolvimento da criatividade em *workshops* através de um processo específico de elicitação de requisitos [9].
- **L. Mich:** a contribuição principal de Mich é a técnica EPMcreate, principalmente descrita em [10]. Uma técnica pragmática com forte apoio da psicologia, adaptada a partir da Técnica EPM.
- **L. Nguyen e outros:** neste trabalho a contribuição está em estudos que apresentam uma visão mais filosófica sobre criatividade na engenharia de requisitos.
- **K. Schimd e outros:** para este trabalho o tema principal de sua propostas é o projeto idSpace, que tem como objetivo desenvolver uma plataforma para a inovação de produtos que suporta a colaboração entre equipes distribuídas.

Segundo Lemos [7] o uso de técnicas que estimulam o pensamento criativo potencializam a elicitação de requisitos mais inovadores, entretanto, alerta que as estratégias de pensamento criativo devem estar plenamente integradas aos atuais processos, métodos e ferramentas de ER. Conforme relatado por Maiden e outros [11], processos de requisitos podem ser aprimorados para produzir resultados mais úteis e inovadores quando modificados sob a ótica da criatividade. Neste sentido alguns pesquisadores desenvolveram ferramentas de software para apoiar especificamente o pensamento criativo durante os processos de elicitação de requisitos [12].

2.2 Criatividade Combinacional

Combinar ideias existentes é uma das técnicas disponíveis na área da criatividade denominada criatividade combinacional. Esta técnica tipicamente requer uma rica fonte de dados e a habilidade de combiná-las de diferentes formas [4]. É possível fazer uma variação de elementos fixos e/ou aleatórios, no qual um

elemento do problema (o elemento fixo) é combinado com um estímulo que é selecionado aleatoriamente obtido através de uma fonte geral de informações como fontes extraídas da *internet* ou outras fontes de problemas específicos como o modelo de domínio [11].

2.3 Processamento da Linguagem Natural

O Processamento de Linguagem Natural (PLN) consiste no desenvolvimento de modelos computacionais que lidam com problemas relacionados à automação da interpretação e da geração da língua humana em diversas aplicações (e.g. tradução e interpretação de textos, extração e recuperação de informações em documentos, sumarização automática de textos, categorização textual) [13].

Conforme Covington e outros [14], a pesquisa em PLN está voltada, essencialmente, a três aspectos da comunicação em língua natural: (i) Som: fonologia; (ii) Estrutura: morfologia e sintaxe; (iii) Significado: semântica e pragmática.

Para Jurafsky [15], o aspecto estrutural da comunicação (morfologia e sintaxe) tem as seguintes definições: morfologia é o estudo da forma como as palavras são construídas a partir da menor partícula significativa da língua e a sintaxe diz respeito ao estudo das relações estruturais entre as palavras, sendo a análise semântica o aspecto da linguagem que permite o processamento de sentenças com foco em seu significado.

Ainda segundo Lima [16] qualquer aplicação de PLN, que apresente um certo grau de sofisticação, necessita de processamento sintático. A construção de gramáticas para esta tarefa tem sido facilitada com o a utilização de modelos estatísticos. Grandes bancos sintáticos como *Penn TreeBank*, composto por notícias do *Wall Street Journal*, permitem que analisadores sintáticos automáticos sejam construídos apresentando resultados com precisão surpreendente (em torno de 92 por cento) para língua inglesa [17].

Para o português brasileiro, um banco de dados com milhões de palavras seria necessário para construção de ferramentas básicas como etiquetadores POS (*Part-Of-Speech*) e *parsers* eficientes, como o que foi construído para o *Penn TreeBank*. Ainda há uma distância muito grande entre os recursos disponíveis para o português e o que hoje existe para língua inglesa [16].

2.4 Topic Modeling

Os cientistas precisam de novas ferramentas para explorar e navegar em grandes coleções de textos dispostos na literatura acadêmica. Graças a organizações como JSTOR³, que digitalizam e indexam os arquivos originais de muitas revistas, os cientistas modernos podem procurar nas bibliotecas digitais documentos que abrangem centenas de anos [18]. Visando o aumento da produtividade, diante de um volume tão grande de dados textuais, os cientistas atuais necessitam de uma forma mais estruturada de manipulação desse conjunto de dados: encontrar

³ <http://www.jstor.org/>

artigos de interesse semelhantes, e explorar o conjunto de textos do seu interesse através dos temas subjacentes, seria uma forma de atingir esse objetivo.

O *Topic Modeling* oferece um conjunto de algoritmos para descobrir estrutura temática escondida em grandes volumes de textos. Os resultados dos algoritmos de *topic modeling* podem ser usados para resumir, visualizar, explorar e teorizar sobre um grande conjunto de dados textuais [19].

No *Topic Modeling* os algoritmos recebem um grande volume de textos como entrada e descobre um conjunto de tópicos/temas recorrentes que são discutidos nos textos de entrada e o grau em que cada documento exhibe esses tópicos. Os tópicos, resultantes do processamento, possibilitam uma maneira resumida de representar os documentos que é útil para navegação e compreensão do conjunto de textos sob diversos aspectos.

2.5 *Part-Of-Speech Tagging*

Classes de palavras ou partes do discursos (*part-of-speech*) é o nome dado a conjuntos abertos de palavras, definidos a partir de propriedades, funções semânticas ou gramaticais [20]. Para Jurafsky [15], a classificação de uma palavra no discurso dá acesso a uma quantidade significativa de informações, não só sobre a própria palavra, mas também sobre seus vizinhos. Esta afirmação é claramente verdadeira não só para as principais classes de palavras (verbos e substantivos), mas também para classes com definições mais finas, como os pronomes possessivos.

Part-Of-Speech Tagging (*POS Tagging*) é o processo de marcação de palavras e caracteres de pontuação em um texto com rótulos que representam as classes de palavras [21]. Apesar dos problemas enfrentados pelas abordagens de *POS tagging*, principalmente os decorrentes da baixa catalogação em bancos de dados específicos e a elevada porcentagem de classificação ambígua das palavras. Segundo Saharia [21], durante as duas últimas décadas, várias abordagens tem sido desenvolvidas para solucioná-los. As abordagens principais podem ser supervisionadas ou não supervisionadas. Ambas são ainda subdivididas em três subtipos; podem ser baseadas em regras, com base estocástica ou em rede neural.

3 Estratégia Proposta

A figura 1 ilustra a estratégia proposta neste trabalho para a elicitación de requisitos utilizando a perspectiva da criatividade combinacional.

O processo para geração das palavras utilizadas para elicitación de requisitos na nossa estratégia consiste nas seguintes atividades:

Extrair textos da documentação do sistema: Nesta primeira atividade, nosso objetivo é identificar os textos de documentação do sistema que sejam relevantes para o processo. Para isso, não fazem parte do corpo dos textos selecionados, imagens, tabelas, permissões de acesso, caminho de menus, diagramas, fluxogramas e demais informações que não sejam referências textuais explícitas das funcionalidades e requisitos do sistema. A entrada de dados nesta tarefa

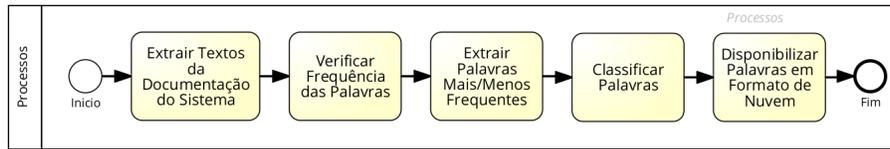


Figura 1. Estratégia Proposta para Elicitação de Requisitos Utilizando Criatividade Combinacional

(documentação do sistema) é informada manualmente pelo Engenheiro de Requisitos que separa os arquivos de documentação a serem processados. O artefato de saída desta tarefa, são arquivos processados e tratados contendo apenas textos referentes às funcionalidades e requisitos do sistema.

Verificar a frequência das palavras: De posse do artefato gerado na atividade anterior, nossa proposta é unificar todos os textos em um só documento e a partir desta nova fonte de informações dispor todas as palavras por ordem de repetição, das mais frequentes às menos frequentes.

Extrair as palavras mais e menos frequentes: Neste processo, nosso objetivo é elencar as n palavras mais e menos frequentes. Onde n é a quantidade definida pelo Engenheiro de Requisitos de acordo com sua necessidade. Uma quantidade muito grande de palavras pode fazer com que o *stakeholder* forme qualquer combinação possível que lhe vier a mente. Em compensação, uma quantidade muito pequena de palavras vai limitar as combinações possíveis de ideias, dificultando assim o processo de geração de novos requisitos.

Classificar as palavras selecionadas: Com base nas palavras selecionadas, necessitamos etiquetá-las segundo a sua classificação gramatical, a saber: Substantivo, Artigo, Adjetivo, Numeral, Pronome, Verbo, Advérbio, Preposição, Conjunção e Interjeição. Em nosso processo, após a classificação, só iremos considerar verbos e substantivos. Palavras pertencentes as demais classes gramaticais serão desconsideradas.

Disponibilizar palavras em formato de nuvem: Por fim, nossa estratégia disponibiliza as palavras selecionadas em formato de nuvem. No nosso caso, duas nuvens são formadas. A primeira constituída por substantivos e a segunda constituída por verbos.

Diante do exposto, segue a sequência de procedimentos que foram adotados para o alcance dos objetivos deste processo.

Com base nisto iniciamos a pesquisa por ferramentas, em Python, que nos auxiliassem no processamento da linguagem natural. Nosso objetivo, nesta fase, era descobrir a estrutura temática por traz de grandes textos (utilizando *Topic Modeling*) além de ordenar suas palavras com base na frequência de repetição. Para isto, utilizamos o *Natural Language Toolkit* (NLTK). O NLTK é um conjunto de módulos, dados e tutoriais de apoio à pesquisa e ensino em linguística computacional e processamento de linguagem natural, escrito em Python e distribuído sob a licença open source GPL [22].

Desta maneira, nossa estratégia foi implementar uma ferramenta que processasse arquivos de texto contendo a documentação de determinado sistema, utilizando NLTK, obtendo ao final do processamento uma lista ordenada das palavras que mais se repetiam nos textos.

Na fase seguinte, de posse da lista de palavras mais frequentes, necessitávamos etiquetar as palavras resultantes segundo a sua classificação morfológica (utilizando *POS Tagging*). No nosso caso, mais especificamente para língua portuguesa. Neste sentido, mesmo o NLTK possuindo uma biblioteca (*MacMorpho Tagged Corpus*) com mais de um milhão de palavras catalogadas a partir de textos jornalísticos (extraídos do jornal Folha de São Paulo) entendemos que a base poderia estar desatualizada uma vez que foi construída em 1994. Além disso, o *MacMorpho Tagged Corpus* não dá suporte a lematização que é o processo de reduzir uma palavra para sua forma-raiz. Assim, palavras no plural após o processo de lematização retornam ao singular e palavras flexionadas verbalmente retornam ao infinitivo, por exemplo.

Desta forma, utilizamos a ferramenta *TreeTagger* desenvolvida por Schmid [23]. Esta ferramenta permite rotular palavras do texto segundo sua classe gramatical, além de realizar a lematização das mesmas. Para que a ferramenta criada por Schmid funcionasse em língua portuguesa, foi necessário utilizar os arquivos de parâmetro criados por Gamallo [24].

Uma vez que as palavras estavam ordenadas e categorizadas segundo sua morfologia, nossa proposta era exibir as n palavras mais frequentes e as n palavras pouco frequentes, agrupadas em forma de “nuvem de palavras” e separadas em substantivos e verbos. O objetivo era que os participantes criassem novas funcionalidades para o sistema em questão, utilizando pelo menos uma palavra de cada nuvem.

4 Estudo de Caso

Com o escopo da proposta definida, escolhemos nossa fonte de dados inicial como sendo a documentação de requisitos do módulo ensino Graduação do Sistema Integrado de Atividades Acadêmicas (SIGAA) desenvolvido na UFRN e utilizados em diversas instituições brasileiras. Este módulo existe para atender as demandas da Pró-reitoria de Graduação, dos departamentos, dos cursos de Graduação, dos docentes e discentes da universidade objetivando auxiliar às atividades intrínsecas a Graduação sob todos os aspectos.

4.1 Aplicação da Estratégia

O referido sistema está documentado utilizando uma versão *Open Source* da *Wiki*, denominado *DokuWiki*. Esta versão é altamente versátil e não requer um banco de dados para armazenar suas informações [25]. Para obter essa documentação, configuramos nosso framework para ler uma lista de links, referentes ao módulo Graduação, remover imagens, tabelas, menus e informações de perfis

de acesso e armazenar separadamente as informações de cada página em um arquivo. Esta atividade foi responsável por gerar a fonte de dados inicial.

Para processar a fonte de dados inicial, realizamos os seguintes passos:

Remoção de palavras: Utilizamos a lista de *stop words* do pacote NLTK em português, de forma que algumas palavras foram removidas do processamento inicial. *Stop words* são palavras consideradas irrelevantes para o contexto do resultado de uma pesquisa. Como exemplo, podemos citar: artigos definidos (o, a, os, as) e indefinidos (um, umas, uns, umas), algumas preposições (de, para, por, em, sobre), pronomes (este, esta, isto, etc) e alguns verbos comuns (estar, haver e suas flexões verbais, por exemplo). Além disso, removemos números e palavras com duas ou menos letras. Tais como: “UF”, “OK”, “nº”, “dá”, “04”, “03”, “i”, “ii”.

Após a remoção das palavras contidas na lista de *stop words*, executamos o método *most_common* contido no pacote NLTK, que nos retornou uma lista de todas as palavras dos textos processados ordenadas das mais frequentes para as menos frequentes.

Através da execução dos testes iniciais foi possível observar que as palavras mais comuns eram palavras “viciadas”. Esse fato foi constatado após a análise do texto original (fonte), uma vez que foi possível observar que algumas frases contidas nos textos seguiam um padrão de repetição. Exemplo disto pode ser verificado em trecho dos arquivos processados: “Caso desista de realizar a **operação clique** em ...” ou “O **usuário** deverá acessar **sistema » graduação** ...”. Estas frases estavam em várias partes dos textos processados. Dessa forma, palavras como “operação”, “clique”, “usuário”, “sistema”, “graduação” apareciam como as mais frequentes. Por outro lado, as palavras com menos frequência (apenas uma aparição em todo o texto) estavam com a grafia errada ou eram apenas erro de digitação. Exemplo destas palavras são: “excluilo”, “com0088”, “23456m123”, “dançacchla”, “estudentisdae”, dentre outras.

Através da observação empírica destes fatos, que poderiam comprometer o resultado da pesquisa, optamos por desconsiderar as 10 primeiras palavras mais frequentes. Para resolvermos o problema das palavras menos comuns, resolvemos calcular a média das palavras mais frequentes e a partir desse valor médio de repetição extraímos as palavras que consideramos menos frequentes.

Para realizar esse cálculo, fizemos uma média aritmética simples onde dividimos o total de palavras válidas (ou seja, após o processamento das remoções) pelo total de palavras únicas (ou seja, sem repetição).

A tabela 1 exhibe um resumo da quantidade de palavras após a remoção de palavras. A tabela 2 exhibe um resumo do conjunto de palavras utilizadas após o processamento.

Ao todo foram processados 291 arquivos referentes à documentação textual de requisitos e funcionalidades do módulo Graduação do SIGAA, totalizando 79.928 palavras. Após a remoção das *stop words* e das palavras com 2 ou menos letras, restou um único arquivo com 48.131 palavras. Para obter a lista de palavras com mais frequência, desconsideramos as 10 primeiras palavras (L01) e recuperamos as 35 primeiras palavras a partir daí (L02). Para obter a lista de palavras com

Tabela 1. Resumo da quantidade de palavras processadas

#	Descrição	Quantidade
a	Total de palavras no texto inicial	79.928
b	<i>Stop words</i>	206
c	Total de palavras após remoção das <i>stop words</i>	50.207
d	Total de palavras após remoção de palavras com 1 ou 2 letras	48.131
e	Total de palavras únicas após processamento	2.671
f	Média de repetições das palavras (d/e)	18

Tabela 2. Resumo do aproveitamento das palavras

Lista	Ordem	Utilização
L01	Da 1 ^a a 10 ^a palavra	Primeiras palavras (desconsideradas)
L02	Da 11 ^a a 45 ^a palavra	Palavras utilizadas, classificadas como mais frequentes
L03	Da 46 ^a a 358 ^a palavra	Palavras desconsideradas
L04	Da 359 ^a a 393 ^a palavra	Palavras utilizadas, classificadas como menos frequentes
L05	Da 394 ^a a 2671 ^a palavra	Demais palavras desconsideradas

menos frequência, recuperamos as 35 palavras que apareciam no mínimo 18 vezes (L04).

Após a obtenção destas duas listas, juntamos todas as palavras recuperadas e executamos a ferramenta TreeTagger, responsável por etiquetar as palavras definindo sua classificação segundo a língua portuguesa: Substantivos, adjetivos, verbos, etc. Com o recurso da lematização, palavras previamente classificadas e divergentes em grau, gênero ou número, foram unificadas. Por exemplo: “selecionar” e “selecione” tornaram-se “selecionar”, assim como “módulos” e “módulo” tornaram-se “módulo”. Assim, o resultado deste processamento gerou duas novas listas a saber: substantivos com 30 palavras e verbos com 24 palavras (Figura 2).

Com base no resultado obtido com o processamento dos textos, publicamos na internet um formulário no Google Drive⁴ que pedia para os respondentes criar de uma a três frases simples e resumidas que descrevessem uma nova funcionalidade para o sistema acadêmico de sua instituição de ensino. Os respondentes deveriam visualizar os dois conjuntos de palavras (substantivos e verbos) que estavam disponibilizados em formato de nuvem e utilizar pelo menos uma palavra de cada nuvem para propor de uma a três novos requisitos.

O link foi disponibilizado para uma turma de 13 alunos de um curso no bacharelado em Engenharia de Software da UFRN, dos quais 11 responderam.

⁴ <http://goo.gl/forms/kekLVFJqhA>



Figura 2. Nuvem de palavras exibidas no formulário de pesquisa

4.2 Discussão

A taxa de resposta dos questionários foi de 84,6%. Entre os estudantes respondentes, pouco mais de 81% tinham entre 18 e 24 anos e apesar de terem pouca experiência com elaboração de requisitos, a maioria deles tinham conhecimento satisfatório do domínio. O uso frequente do sistema acadêmico para o qual os novos requisitos foram sugeridos associado à possibilidade de solucionar algumas de suas necessidades reais, fizeram dos estudantes um perfil de *stakeholder* adequado para execução do experimento.

Os requisitos elaborados pelos respondentes foram classificados nos seguintes termos: requisitos válidos, inválidos, novos e comuns.

Requisito válido: um requisito foi classificado como válido quando a frase elaborada satisfazia as condições estabelecidas no experimento, quais sejam: conter, pelo menos, um verbo e um substantivo das nuvens de palavras apresentadas.

Requisito inválido: um requisito foi classificado como inválido quando, em sua composição, não constavam pelo menos um verbo e um substantivo das nuvens de palavras geradas.

Requisito Novo: é todo e qualquer requisito válido que não tenha sido considerado durante a elaboração do SIGAA.

Requisito Comum: é todo requisito válido que já tenha sido considerado pela equipe de desenvolvedores do SIGAA e que se encontrava disponível na versão mais atual do sistema.

Nós analisamos cada requisito válido individualmente e para a sua classificação como novo ou comum, uma pesquisa exploratória foi realizada no SIGAA através de um perfil de usuário compatível com o perfil dos alunos respondentes. A pesquisa buscou por funcionalidades compatíveis com as sugeridas pelos respondentes que já estavam implementadas no sistema.

A Tabela 3 mostra o sumário de nossa análise e como podemos ver, a maioria dos requisitos foram classificados como inovadores (44%) para o SIGAA.

Tabela 3. Resumo da classificação dos requisitos

Classificação dos Requisitos	Quantidade	Percentual
Novos	11	44%
Comuns	9	36%
Inválidos	5	20%

Na Tabela 4, são apresentados alguns exemplos de requisitos sugeridos e a classificação realizada. Os substantivos e verbos destacados indicam sua presença nas nuvens de palavras fornecidas.

Tabela 4. Exemplos de requisitos sugeridos

#	Descrição do Requisito Proposto	Validade	Inovação
RQ01	O discente será notificado quando uma nota for fornecida.	Válido	Novo
RQ02	O aluno realiza matricula .	Válido	Comum
RQ03	Calcular a quantidade de faltas restantes.	Inválido	-

Como dito anteriormente, nossa estratégia visa auxiliar o Engenheiro de Requisito principalmente na fase de Elicitação de Requisitos. No entanto, com o intuito de permitir uma maior flexibilização nas configurações do *framework* proposto, detectamos que algumas variações poderiam ser aplicadas na nossa estratégia.

1. **Disponibilização das palavras:** O formato de nuvem é um dos diferenciais da nossa estratégia quando comparado com a proposta de Bhowmik [6]. Temos a pretensão de fazer algumas variações nas Nuvens de Palavras, de forma a dar destaque às palavras menos frequentes e remover os destaques das palavras mais frequentes. Outra abordagem a ser considerada é substituir a nuvem de palavras por listas e tabelas, mas sempre destacando algumas palavras em detrimento de outras.
2. **Ambiente para realização do experimento:** Nosso experimento foi realizado através de formulário disponível na internet. Os participantes puderam acessar o formulário de qualquer lugar e até mesmo através de *smartphones* ou *tablets*. Uma variação desta estratégia seria fazer uma reunião de *brainstorming* com os principais participantes do projeto e discutir as combinações de palavras propostas pelos mesmos gerando ainda mais ideias.
3. **Combinações de palavras do glossário:** Na nossa estratégia, não consideramos palavras do glossário. Tecnicamente estas palavras tem grande

representatividade para o domínio em questão. Uma variação seria permitir a utilização destas palavras na combinação com as demais já selecionadas. Ou até mesmo, remover as palavras mais frequentes e combinar as do glosário com as menos frequentes.

4. **Fonte de dados externa adicional:** Como a criatividade combinacional sugere que um estímulo aleatório pode ser obtido através de uma fonte geral de dados, como a *internet*, temos a pretensão de utilizar redes sociais para extrair uma parte dos dados e combinar com os dados da fonte de domínio.

4.3 Limitações

Como relatado anteriormente, apesar de termos configurado nosso algoritmo para geração de 35 substantivos e 35 verbos, o resultado final contabilizou apenas 30 substantivos e 24 verbos, devido ao processo de lematização ter ocorrido após o processo de verificação de repetição das palavras. Essa redução, apesar de não ter exercido forte influência na quantidade de palavras (os substantivos tiveram uma redução de 15% das palavras e verbos uma redução de 31%), poderia ser ainda maior a depender da estruturação do texto e da quantidade de palavras selecionadas na configuração inicial.

Uma limitação da nossa estratégia refere-se à ausência de um processamento para validar minimamente o envio das sugestões de novos requisitos. Esta validação deveria verificar se as palavras inseridas nos requisitos propostos fazem parte da nuvem de palavras disponibilizadas. Além disso, poderia ser validado também, através de processamento de linguagem natural, se a sugestão é positiva ou não para o sistema em questão. Neste sentido, esse processamento seria uma variação do processo proposto em [26].

Outra limitação do estudo diz respeito à ausência de uma metodologia que avalie o grau de inovação dos requisitos descobertos. Algumas técnicas tem sido propostas para medir a inovação de um requisito. Ritchie [27] postulou um conjunto de critérios formais que poderiam ser aplicados para avaliar o comportamento criativo de programas de software.

5 Conclusões e Trabalhos Futuros

Neste trabalho, contribuímos com o fornecimento de uma estratégia para elicitacão de requisitos através da perspectiva da criatividade combinacional como em [6]. O estudo apresenta resultados satisfatórios na descoberta de novos requisitos para o Sistema Integrado de Atividades Acadêmicas - SIGAA, onde 44% das ideias propostas pelos usuários participantes do experimento foram classificadas como originais e relevantes para o sistema. Além de facilitar a descoberta de novos requisitos pelos *stakeholders*, a proposta apresentada estimula o pensamento criativo nos desenvolvedores, refletindo em melhorias no aspecto criativo do sistema como um todo.

Como trabalhos futuros, podemos apontar:

Automatizar completamente o *framework*, disponibilizando-o como ferramenta *online*, afim de que todas as fases da nossa estratégia possam ser realizadas e configuradas pelo Engenheiro de Requisitos.

Alterar a ordem de processamento da fonte de dados, de forma que o algoritmo de *POS Tag* seja executado antes do *Topic Modeling*. Por consequência, pegaremos as palavras lematizadas para o processamento das palavras mais frequentes.

Possibilitar mais de um tipo de visualização das palavras, além do formato de nuvem já explorado. Neste sentido, testaríamos listas de palavras e tabelas por exemplo.

Realizar novos experimentos com sistemas fora da área acadêmica.

Referências Bibliográficas

- [1] Anderson Belgamo and Luiz Eduardo Galvão Martins. Estudo comparativo sobre as técnicas de elicitação de requisitos do software. In *XX Congresso Brasileiro da Sociedade Brasileira de Computação (SBC), Curitiba-Paraná, 2000*.
- [2] Edinelson Aparecido Batista and Ariadne Maria Brito Rizzoni Carvalho. Uma taxonomia facetada para técnicas de elicitação de requisitos. In *WER*, pages 48–62, 2003.
- [3] Robert J Sternberg. *Handbook of creativity*. Cambridge University Press, 1999.
- [4] Margaret A Boden. *The creative mind: Myths and mechanisms*. Psychology Press, 2004.
- [5] L Pennell and NAM Maiden. Creating requirements—techniques and experiences in the policing domain. In *Proceedings REFSQ'2003 Workshop*. Citeseer, 2003.
- [6] Tanmay Bhowmik, Nan Niu, Anas Mahmoud, and Juha Savolainen. Automated support for combinational creativity in requirements engineering. In *Requirements Engineering Conference (RE), 2014 IEEE 22nd International*, pages 243–252. IEEE, 2014.
- [7] João Lemos, Carina Alves, Leticia Duboc, and Genaina Nunes Rodrigues. A systematic mapping study on creativity in requirements engineering. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 1083–1088. ACM, 2012.
- [8] Neil Maiden and Alexis Gizikis. Where do requirements come from? *Software, IEEE*, 18(5):10–12, 2001.
- [9] James Robertson. Eureka! why analysts should invent requirements. *IEEE Software*, 19(4):20–22, 2002.
- [10] Luisa Mich, Cinzia Anesi, and Daniel M Berry. Applying a pragmatics-based creativity-fostering technique to requirements elicitation. *Requirements Engineering*, 10(4):262–275, 2005.
- [11] Neil Maiden, Sara Jones, Kristine Karlsen, Roger Neill, Konstantinos Zachos, and Alastair Milne. Requirements engineering as creative problem solving: A research agenda for idea finding. In *Requirements Engineering Conference (RE), 2010 18th IEEE International*, pages 57–66. IEEE, 2010.
- [12] Inger Kristine Karlsen, Neil Maiden, and Andruud Kerne. Inventing requirements with creativity support tools. In *Requirements Engineering: Foundation for Software Quality*, pages 162–174. Springer, 2009.

- [13] Comissão Especial de Processamento de Linguagem Natural. Sbc, comissões especiais, processamento de linguagem natural @ONLINE, 2009 (acessado em 20 de Dezembro de 2014).
- [14] Michael A. Covington, Donald Nute, and André Vellino. *Prolog Programming in Depth*. Scott, Foresman & Co., Glenview, IL, USA, 1987.
- [15] Dan Jurafsky and James H Martin. *Speech & language processing*. Pearson Education India, 2000.
- [16] Vera Lúcia Strube de Lima, Maria das Graças Volpe Nunes, and Renata Vieira. Desafios do processamento de línguas naturais. *SEMISH-Seminário Integrado de Software e Hardware*, 34:1, 2007.
- [17] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993.
- [18] David M Blei and John D Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10:71, 2009.
- [19] D. Blei. Topic modeling and digital humanities. *Journal of Digital Humanities*, 1(2):8–11, 2012.
- [20] Margarida Basilio. *Formação e classes de palavras no português Brasil*. Editora Contexto, 2008.
- [21] Navanath Saharia, Dhruvajyoti Das, Utpal Sharma, and Jugal Kalita. Part of speech tagger for assamese text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 33–36, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [22] Nianwen Xue. Steven bird, evan klein and edward loper. natural language processing with python. o'reilly media, inc. 2009. isbn: 978-0-596-51649-9. *Natural Language Engineering*, 17(03):419–424, 2011.
- [23] Helmut Schmid. Part-of-speech tagging with neural networks. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 172–176. Association for Computational Linguistics, 1994.
- [24] Pablo Gamallo and Marcos Garcia. Freeling e treetagger: um estudo comparativo no âmbito do português.
- [25] Andreas Gohr. Dokuwiki. URL: <http://dokuwiki.org> (visited on 15/11/2014), 2010.
- [26] Emitza Guzman and Walid Maalej. How do users like this feature? a fine grained sentiment analysis of app reviews. In *Requirements Engineering Conference (RE), 2014 IEEE 22nd International*, pages 153–162. IEEE, 2014.
- [27] Graeme Ritchie. Assessing creativity. In *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in Arts and Science*, pages 3–11, 2001.