



**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

ALAN VINICIUS CEZAR ENSINA

**Estudo comparativo de tecnologias de processamento de
linguagem natural para avaliação de histórias de usuário**

Florianópolis
2022

ALAN VINICIUS CEZAR ENSINA

**Estudo comparativo de tecnologias de processamento de
linguagem natural para avaliação de histórias de usuário**

Trabalho de conclusão de curso submetido ao curso de Sistemas
de Informação para a obtenção do grau de Bacharel em Sistemas
de Informação pela Universidade Federal de Santa Catarina –
UFSC

Orientadora: Prof^a Dr^a Fabiane Barreto Vavassori Benitti

Florianópolis
2022

ALAN VINICIUS CEZAR ENSINA

Estudo comparativo de tecnologias de processamento de linguagem natural para avaliação de histórias de usuário

Este Trabalho de conclusão de curso foi julgado aprovado para a obtenção do Título de “Bacharel em Sistemas de Informação” e aprovado em sua forma final pelo curso de Sistemas de Informação.

Florianópolis, XX de dezembro de 2022

Profº Drº. Álvaro Junio Pereira Franco
Coordenador do Curso

Banca examinadora:

Profº Drª Fabiane Barreto Vavassori Benitti
Orientadora

XXX

XXX

RESUMO

Histórias de usuário são as representações das necessidades de um usuário e são utilizadas para facilitar o entendimento entre a equipe de negócios e a equipe de desenvolvimento para obter um maior acerto no desenvolvimento do produto com base na especificação. Porém, devido ao fato de serem escritas de maneira simples e curtas, diversas vezes podem causar dúvidas no momento da implementação. Sendo assim, é necessário encontrar uma forma de automatizar a avaliação dessas histórias afim de obter uma maior completude, uniformidade e consistência. Se tratando de automatização, o Processamento de Linguagem Natural (PLN) é uma subárea da inteligência artificial capaz de compreender automaticamente línguas humanas naturais capaz de automatizar diversos processos, porém devido ao alto número de tecnologias de PLN presente hoje no mercado, ainda é necessário compará-las para que seja possível aferir qual tecnologia possui, por exemplo, uma maior exatidão em seus processamentos, melhor performance e qual é a mais adequada no contexto de histórias de usuário. O presente trabalho pretende realizar uma análise comparativa entre soluções de PLN para a avaliação de histórias de usuário.

Palavras-chave: engenharia de software, histórias de usuário, processamento de linguagem natural, PLN

ABSTRACT

User Stories are representations of a user's needs and are used to help the understanding between the business team and the development team to achieve greater accuracy in product development based on the specification. However, due to the fact that they are written in a simple and short way, they can often cause doubts at the time of implementation. Therefore, it is necessary to find a way to automate the evaluation of these stories in order to obtain greater completeness, uniformity and consistency. When it comes to automation, Natural Language Processing (NLP) is a subarea of artificial intelligence capable of automatically understanding natural human languages capable of automating several processes, but due to the high number of NLP technologies present on the market, it is still necessary to compare them so that it is possible to assess which technology has, for example, greater accuracy in its processing, better performance and which is the most appropriate in the context of user stories. The present work intends to carry out a comparative analysis between NLP solutions for the evaluation of user stories.

Keywords: software engineering, user stories, natural language processing, NLP

LISTA DE FIGURAS

Figura 1 XXX XX

LISTA DE TABELAS

Tabela 1 XXX XX

LISTA DE ABREVIATURAS E SIGLAS

PLN	Processamento de Linguagem Natural	XX
-----	--	----

SUMÁRIO

1. INTRODUÇÃO

“Tempo é dinheiro” (FRANKLIN, 1748) famosa frase dita por Benjamin Franklin na metade do século 18 ainda ecoa na cabeça de muitos seres humanos. Em busca de mais tempo as pessoas procuram então otimizar suas tarefas. Uma forma de otimizar as tarefas é a criação de automações. As automações buscam por uma melhor produtividade, redução de custos e maior tempo livre para se concentrar em outras tarefas que não podem ser automatizadas. Silva (2019) define que “... automação é um dos processos mais utilizados para a facilitação de inserção dos recursos tecnológicos. Através dessa tecnologia, são utilizadas ferramentas para soluções tecnológicas com o objetivo de otimizar e tornar simples os processos internos, além de diminuir custos operacionais.”

Um grande exemplo disso são as assistentes virtuais, como por exemplo a Alexa da Amazon, a Siri da Apple e o Google Home do Google. Esses assistentes virtuais são capazes de realizar diversas tarefas através de um simples comando de voz. Essa interação entre seres humanos e máquinas está cada vez mais presente nos sistemas, mas para que isso seja possível, é utilizado o Processamento de Linguagem Natural - PLN (RACKSPACE TECHNOLOGY, 2020).

Johnson (2021) define o PLN sendo um ramo dentro da Inteligência Artificial responsável em fazer com que as máquinas possam compreender a linguagem dos seres humanos, ou seja, o PLN funciona como um tradutor, permitindo assim que as tecnologias possam entender seus usuários, mesmo eles utilizando a linguagem natural.

O PLN também está presente em outras plataformas além das assistentes virtuais. Por exemplo, ele auxilia em sites de busca realizando interpretações entre o que o usuário digita com conteúdos de sites que poderão ser exibidos. Também está presente no auto-completar em plataformas de busca, onde sugestões automáticas são exibidas na tela no momento em que o usuário está digitando. Chatbots, que são utilizados por empresas para se comunicar com seus clientes, também fazem uso do PLN realizando a “tradução” do que o cliente deseja com possíveis soluções das quais as empresas podem oferecer (TAKE BLIP, 2019).

Para que seja possível criar sistemas voltados a automações, é necessário levantar os requisitos que esse sistema irá possuir. Em engenharia de requisitos, a etapa responsável para o levantamento dessas informações é a elicitacão. Para Thayer (1997), a elicitacão de requisitos é o processo em que os clientes e usuários são questionados pela equipe de desenvolvimento a falarem o quê espera como funcionalidades no sistema que será desenvolvido. Nessa etapa de elicitacão serão definidas as exigências, os recursos, os objetivos e as utilidades que o sistema deve cumprir.

Segundo Sommerville(2011, pág. 57):

Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições de seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma funcionalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações. O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado engenharia de requisitos.

A especificacão de requisitos no desenvolvimento ágil pode ser feito por meio de histórias de usuário (User Stories). Através delas, o usuário utiliza de uma abordagem de escrever sobre os requisitos, tudo isso por meio de uma ou duas frases escritas na perspectiva de quem deseja o recurso/funcionalidade.

Para Cohn (2009, pág. 4), “uma história de usuário descreve a funcionalidade que será valiosa para um usuário ou comprador de um sistema ou software”. Já Rehkopf (2020) define histórias de usuário como “uma explicacão informal e geral sobre um recurso de software escrita a partir da perspectiva do usuário final. Seu objetivo é articular como um recurso de software pode gerar valor para o cliente.”

As técnicas de PLN também podem oferecer diversas vantagens para melhorar a qualidade das histórias de usuário. Segundo Raharjana, Siahaan e Fatichah (2021):

As técnicas de processamento de linguagem natural (PLN) oferecem vantagens potenciais para melhorar a qualidade das histórias de usuários. O PLN pode ser usado para analisar ou extrair os dados da história do usuário. Tem sido amplamente utilizado para ajudar no domínio da engenharia de software (por exemplo,

gerenciamento de requisitos de software, extração de atores e ações no documento de requisitos, teste de software, etc.).

1.1 PROBLEMA

Cohn (2009) comenta que ao definir os requisitos de software a comunicação pode ser uma adversidade, pois aqueles que desejam um novo software devem se comunicar com quem irá desenvolvê-lo.

Heck (2014) propõe critérios específicos para avaliar a qualidade em histórias de usuários: completude, uniformidade, consistência e correção. Porém, muitos desses critérios, no entanto, requerem informações complementares que não são capturadas em um texto de história do usuário. Femmer (2013) define o termo *Requirement Smell* como indicador de má qualidade na especificação de requisitos. Femmer (2014) subdivide o *Requirement Smell* em 9 tipos: ambiguidade de advérbios e adjetivos, pronomes vagos, linguagem subjetiva, comparações, superlatividade, afirmações negativas, termos não verificados, *loopholes* (*brechas*) e referências não verificadas.

Dentro do contexto de histórias de usuário, seria possível avaliá-las utilizando soluções de PLN levando em consideração os critérios de qualidade?

Atualmente existem inúmeras soluções utilizadas para o PLN. Parker(2019) em seu artigo cita 12 ferramentas open source em diversas linguagens de programação, como por exemplo Python, Node e Java. Dentre as soluções citadas por Parker (2019), destaca-se a Natural Language Toolkit (NLTK) em Python, por ser a solução com mais recursos disponíveis, capaz de implementar todos os componentes de PLN e oferece suporte a vários idiomas. Outra solução que se destaca é a OpenNLP em Java. É hospedada pela Apache Foundation, ou seja, é fácil integrá-la com outros serviços da Apache. Assim como a NLTK, oferece suporte a vários idiomas e cobre todos os componentes de PLN.

No entanto, ainda é necessário compará-las para que seja possível aferir qual tecnologia possui, por exemplo, uma maior exatidão em seus processamentos e qual possui a melhor performance. Sendo assim, levando

em consideração os critérios de qualidade (completude, uniformidade e consistência), qual a solução mais adequada para o PLN no contexto de histórias de usuário?

Neste sentido, o objetivo central de desse trabalho é realizar um estudo sistemático de comparação entre pequenas soluções utilizando PLN para avaliação de histórias de usuário para avaliação da qualidade de histórias de usuário nos idiomas português e inglês.

1.2 SOLUÇÃO PROPOSTA

Com base no cenário atual, no qual existem diversas tecnologias voltadas para PLN, se faz necessário uma análise comparativa entre essas tecnologias afim de definir qual ou quais tecnologias são mais adequadas para a avaliação de histórias de usuário.

Para que isso seja possível, serão selecionadas algumas tecnologias para que sejam previamente avaliadas em aspectos como por exemplo: documentação, linguagem de programação, conteúdo disponível na internet a respeito da tecnologia (sites, fóruns e *threads* em redes sociais) e o uso atual no mercado. Após feito esse levantamento de dados, as tecnologias que mais se destacarem serão selecionadas como objetos de estudo e serão implementados protótipos voltados a avaliação de histórias de usuário.

Esses protótipos serão avaliados levando em consideração critérios de qualidade em requisitos de software, como por exemplo eficiência, acurácia e funcionalidades. Quanto as histórias de usuário, serão selecionados alguns critérios de qualidade, conforme literatura, para avaliação.

Para além dos critérios de qualidade dos requisitos, pretende-se também avaliar nos protótipos aspectos relacionados a eficiência no processamento para os idiomas inglês e português e também a produtividade da tecnologia.

1.3 OBJETIVOS

1.3.1 OBJETIVO GERAL

Desenvolver um estudo comparativo entre soluções de PLN com o propósito de avaliar qual ou quais tecnologias são mais adequadas para analisar critérios de qualidade em requisitos de software descritos como história de usuário.

1.3.2 OBJETIVOS ESPECÍFICOS

- Analisar e avaliar soluções atuais no mercado, comparando-as dentro dos critérios estabelecidos;
- Implementar dois protótipos voltados para a avaliação de histórias de usuário utilizando as duas soluções de PLN mais bem avaliadas;
- Avaliar os protótipos desenvolvidos

1.4 METODOLOGIA

Metodologia é a estrutura filosófica dentro da qual a pesquisa é conduzida ou a base sobre a qual a pesquisa se baseia (BROWN, 2006). Já O'Leary (2004) descreve a metodologia como a estrutura que está associada a um conjunto particular de suposições paradigmáticas usadas para conduzir a pesquisa.

Sendo assim, dado o contexto de metodologia, o estudo seguirá o modelo científico em camadas (Research Onion) de Saunders (2007), seguindo a forma transversal, indutiva e interpretativa. Seguirá um modelo multimétodo, com procedimento de pesquisa bibliográfica (GIL, 2010), estudo comparativo das tecnologias (FACHIN, 2001), design e prototipação (SOMMERVILLE, 2011) e Goal Question Metric (GQM) (BASILI, CALDIERA, ROMBACH, 1994).

Etapa	Atividades	Métodos	Resultados
Etapa 1 - Síntese da fundamentação teórica	- Sintetizar contexto histórico de processamento de linguagem natural, de histórias de usuário e de critérios de qualidade	Pesquisa bibliográfica (GIL, 2010)	Fundamentação teórica
Etapa 2 - Estudo comparativo	- Pesquisar as tecnologias mais utilizadas - Análise das tecnologias conforme os requisitos preestabelecidos - Definir as duas tecnologias mais promissoras	Estudo comparativo (FACHIN, 2001)	Análise comparativa de potenciais soluções
Etapa 3 - Prototipação	- Implementar protótipos das soluções A e B em português e inglês	- Design e prototipação (SOMMERVILLE, 2011)	Protótipos das tecnologias selecionadas
Etapa 4 - Avaliação comparativa	- Avaliar e comparar os resultados obtidos das tecnologias A e B	- GQM (BASILI et al., 1994)	Avaliação das tecnologias e tabela comparativa

Tabela 01 - Etapas e metodologias aplicadas

2. FUNDAMENTAÇÃO TEÓRICA

Com o objetivo de contextualizar os temas abordados neste estudo, neste capítulo são introduzidos um embasamento teórico sobre histórias de usuário, critérios de qualidade e processamento de linguagem natural.

2.1 HISTÓRIAS DE USUÁRIO

Segundo Francino (2017), o conceito de histórias de usuário foi introduzido pela primeira vez em 1998 na XP (Extreme Programming) comparando-as com Casos de Uso. Com o aumento da popularidade da XP e do Scrum, as histórias de usuário se tornaram uma abordagem muito conhecida para a definição de requisitos.

Uma história de usuário pode ser descrita como uma frase curta e semiestruturada capaz de ilustrar os requisitos de um software na perspectiva do usuário, ou seja, pode ser usada para identificar o desejo do usuário com relação ao produto (RAHARJANA, HARRIS, JUSTITIA. 2020).

Wautelet, et al. (2017) definem que uma história de usuário consiste de quatro elementos:

- **Papel:** comportamento esperado do ator no contexto do problema
- **Objetivo:** condição desejada pelas partes interessadas
- **Tarefa:** obrigação específica que devem ser realizadas a fim de atingir os objetivos
- **Capacidade:** a habilidade dos atores em atingir as metas com base em certas condições ou eventos

Já para Cohn(2009), para facilitar a escrita de histórias de usuário, ele sugere o seguinte template:

“Como <tipo de usuário>, quero <algum objetivo> para que <algum motivo>”

Exemplo:

“Como cliente, quero utilizar a forma de pagamento por pix para que eu possa pagar minha compra.”

Cohn(2009) também afirma que mais importante do que escrever histórias de usuários é a discussão a respeito dela, sendo assim, sugere que

elas devem ser escritas em pequenos papéis ou até mesmo em notas adesivas, para que sejam facilmente expostas em paredes ou murais para facilitar o planejamento e a discussão.

2.2 CRITÉRIOS DE QUALIDADE

Ao realizar as especificações de requisitos de software, o IEEE recomenda 9 características bases para qualidade de requisitos: necessária, apropriada, não ambígua, completa, singular, praticável, verificável, correta e conforme (IEEE Computer Society, 2018).

Se tratando de critérios de qualidade voltados a requisitos de software, Heck e Zaidman (2014) desenvolveram um framework voltado a verificação de requisitos ágeis, neste framework, foram definidos três critérios de qualidade para verificação de alto nível:

Completeness: Todos os elementos necessários na Área de Negócios (*Business Area*) devem estar presentes.

Uniformidade: O estilo dos elementos da Área de Negócios deve ser padronizado.

Consistência e correção. Todos os elementos devem estar em conformidade com a propriedade objeto da certificação.

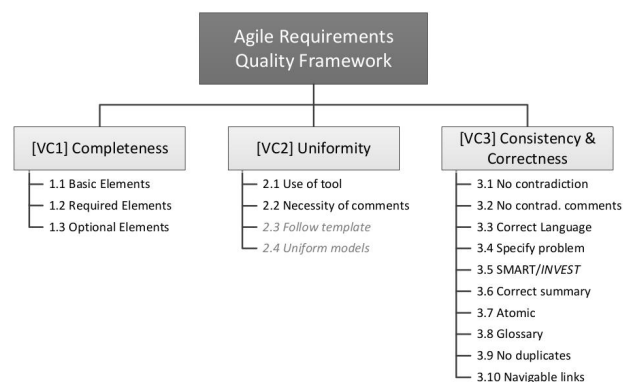


Figura 01: Critérios de qualidade definidos por Heck e Zaidman (2014)

Na Figura 01, é possível observar como cada critério de qualidade é subdividido, os itens em *itálico* são aplicados apenas para histórias de usuário, já o restante, são aplicados tanto para histórias de usuário quanto para solicitações de requisitos de software.

Lucassen, et. al. (2016) em seu estudo desenvolve um framework voltado a avaliação de histórias de usuário utilizando processamento de linguagem natural. Neste framework são utilizados 13 critérios de qualidade a serem validadas, sub-divididas em 3 grupos: sintáticas, semânticas e pragmáticas.



Figura 02: Critérios de qualidade definidos por Lucassen, et. al. (2016)

Se tratando de critérios de qualidade definidos pela IEEE (IEEE Computer Society, 2018), Heck e Zaidman (2014) e Lucassen, et. al. (2016), nota-se que há muitos critérios com o mesmo significado, porém com nomenclaturas diferentes, sendo assim, segue abaixo uma tabela comparativa destacando os critérios definidos por cada autor juntamente com os critérios estabelecidos pela IEEE (IEEE Computer Society, 2018).

IEEE Computer Society (2018)	LUCASSEN et. al. (2016)	Heck e Zaidman (2014)
Necessária		
Apropriada	Sem conflito	
Não ambígua	Não ambígua	
Completa	Completa	Compleitude
Singular	- Única - Independente - Atômica	

Praticável	Estimável	
Verificável		
Correta	Bem formada	Consistência e correção
Conforme	Uniforme	Uniformidade
	Mínima	
	Orientada ao problema	
	Sentença completa	
	Conceito sólido	

Tabela 2 - Critérios de qualidade definidos por autor

Por outro lado, atualmente existem diversos estudos voltados a indicadores de má qualidade na especificação de requisitos, sendo assim, Nascimento et. al (2018) apresenta em seu estudo um mapeamento sistemático de literaturas que investigam a existência de indicadores de má qualidade que podem prejudicar negativamente a compreensão, manutenção e qualidade dos artefatos. Estes indicadores são descritos pelo autor como *Requirement Smells*.

No mapeamento sistemático apresentado por Nascimento et. al. (2018), 41 estudos são analisados desde 2013 onde 9 tipos de *Requirement Smells* são citados. Segue abaixo uma tabela comparativa dos *Requirement Smells* citados e o critério de qualidade oposto definido pela IEEE (IEEE Computer Society, 2018):

Requirement Smells (Nascimento et. al. (2018))	Descrição do Requirement Smell (Nascimento et. al. (2018))	Critérios de qualidade (IEEE Computer Society, 2018)
Advérbios e Adjetivos Ambíguos	Adjetivos e advérbios que causam ambiguidade na compreensão dos requisitos. Exemplo: Se a qualidade for muito baixa , uma falha deve ser gravada na memória de erros.	Não ambígua

Pronomes vagos	São pronomes com relações pouco claras. Exemplo: O software deve implementar serviços para aplicativos, que devem se comunicar com os aplicativos do controlador implantados em outros controladores.	SCompleta
Linguagem subjetiva	São palavras cuja semântica não é objetiva. Exemplos: amigável, fácil de usar, econômico.	Apropriada
Comparações específicas	São advérbios e adjetivos, onde os requisitos expressam uma relação do sistema com outros sistemas específicos. Exemplo: melhor que, maior qualidade.	Singular
Advérbios e adjetivos superlativos	São advérbios e adjetivos, onde os requisitos expressam uma relação do sistema com todos os outros sistemas. Exemplo: melhor desempenho, menor tempo de resposta.	Apropriada
Afirmações negativas	São palavras usadas em funcionalidades que o sistema não deve fornecer, pois podem levar a falta de explicação sobre o comportamento do sistema em tais casos. Exemplo: o sistema não deve aceitar cartões de crédito VISA.	Apropriada
Termos não verificáveis	São palavras difíceis de verificar por oferecer várias possibilidades de execução do sistema. Exemplo: O sistema só pode ser ativado se todos os sensores necessários (...) trabalharem com precisão de medição suficiente.	Verificável
Loopholes	São palavras que possibilitam os stakeholders ignorar as especificações. Exemplos: se possível, conforme apropriado, conforme aplicável.	Conforme
Referências incompletas	São referências que o leitores não conseguem encontrar	Completa

Tabela 3 - Tabela comparativa entre Requirement Smells e critérios de qualidade

2.3 PROCESSAMENTO DE LINGUAGEM NATURAL

Após a segunda guerra mundial, as pessoas notaram a necessidade em traduzir informações de um idioma para outro, sendo assim, esperavam automatizar esse processo através de uma máquina capaz de realizar essas traduções automaticamente, foi então o início dos estudos na área de processamento de linguagem natural (ROBERTS, 2004).

O processamento de linguagem natural (PNL) é uma disciplina que combina linguística, ciência da computação e inteligência artificial para estudar as interações entre sistemas de computador e linguagem natural humana (FERRARIO et. al., 2020).

O uso de PLN está presente em uma variedade de aplicações do mundo real em vários campos, incluindo pesquisa médica, mecanismos de pesquisa e inteligência de negócios (LUTKEVICH, 2021).

○ PLN pode ser dividido em duas abordagens: clássica e estatística. Porém, mesmo com essa divisão não significa necessariamente que uma é superior a outra. Um exemplo disso é que na abordagem estatística requer uma grande quantidade de dados rotulados no idioma desejado, ou seja, é mais viável utilizar quando o conjunto de dados é vasto. Já na abordagem clássica, não há uma necessidade de estar presa a um só idioma, pois há uma sequência de passos pré-definidos e se faz necessário apenas o conhecimento sobre a estrutura do idioma utilizado (BAASCH, 2021).

2.3.1 Abordagem clássica

Tradicionalmente, o processamento de linguagem natural na abordagem clássica tende a ser um processo decomposto em etapas, sendo elas espelhadas em distinções linguísticas teóricas entre sintáticas, semânticas e pragmáticas (INDURKHYA; DAMERAU, 2010).

Durante esse processo o texto é dividido em sentenças onde a sintaxe dos termos são analisadas, buscando produzir uma estrutura mais amigável à análise semântica. Por fim, uma análise pragmática é realizada com o propósito de avaliar a palavra ou sentença possui sentido dentro do contexto (BAASCH, 2021).

Com o conhecimento disponível hoje, este processo foi refinado e decomposto em: tokenização, análise léxica, análise sintática, análise semântica e análise pragmática (BAASCH, 2021).

Segue abaixo uma figura ilustrando as etapas do PLN na abordagem clássica, onde o texto a ser processado é recebido, passa etapa de tokenização, análise léxica, análise sintática, análise semântica e análise pragmática, para que no fim a máquina tenha o entendimento do texto recebido no início do processo.

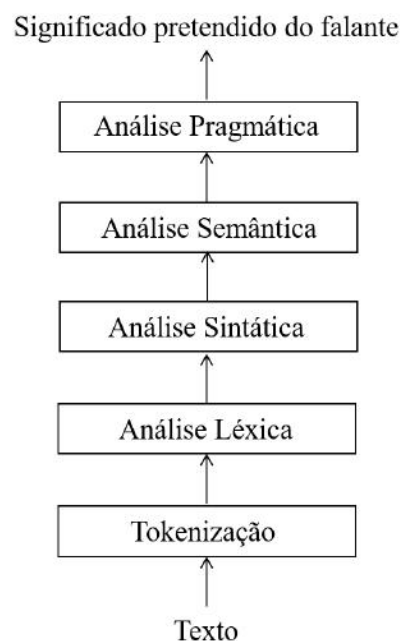


Figura 03: Etapas de análise em um processamento de linguagem natural (INDURKHYA; DAMERAU, 2010)

2.3.1.1 Pré-processamento de texto

Antes de inicializar o processamento do texto, esse texto recebido como entrada deve ser tratado, afim de identificar erros que prejudiquem as análises. O pré-processamento de texto é uma etapa essencial no processamento de linguagem natural, pois as palavras e sentenças serão utilizadas como base das etapas subsequentes (INDURKHYA; DAMERAU, 2010).

Para que a análise seja eficiente, é preciso definir quais serão os caracteres utilizados pelo texto, juntamente das palavras e sentenças. Nesta etapa, deve-se realizar um processo de limpeza que consiste em identificar a

codificação do texto e convertê-la para a codificação que será utilizada, também deve-se remover as imagens presentes no texto, links, tags HTML ou qualquer outro elemento que não traga valor ao texto que será processado (INDURKHYA; DAMERAU, 2010).

2.3.1.1.1 Tokenização



Tokenização é um processo fundamental dentro do PLN. A tokenização é o processo de separar um pedaço de texto em unidades menores, o então chamados *tokens*. Os *tokens* podem ser classificados em três tipos: tokenização de palavras, tokenização de caracteres e tokenização de subpalavras (PAI, 2020).

A maneira mais comum de criar *tokens* é baseada em espaços em branco. Por exemplo, na frase: “Nunca desista”, assumindo o espaço como um delimitador, cada palavra será um *token*, logo são identificados dois *tokens* para esta frase (PAI, 2020).


Exemplo: Deep Learning		
Tokenização por palavra	Tokenização por caracter	Tokenização por subpalavra
Deep, Learning (2 <i>tokens</i>)	D,e,e,p,L,e,a,r,n,i,n,g (12 <i>tokens</i>)	Deep, Learn, ing (3 <i>tokens</i>)

Tabela 04 - Tipos de tokenização e exemplo



Pode-se observar na tabela 04 como os três tipos de tokenização se comporta. Como exemplo foi dado o texto “*Deep Learning*”, onde na tokenização por palavra utilizando o espaço como delimitador, sendo assim foram gerados dois *tokens*. No segundo exemplo, na tokenização por caracter, cada caracter de cada palavra gera um novo *token*, portanto, doze *tokens* foram gerados. No terceiro exemplo, na tokenização por subpalavra, foram gerados três *tokens*, pois ‘*ing*’ juntamente de um verbo no idioma inglês determina o gerúndio de uma ação.

O processo de tokenização pode sofrer algumas dificuldades, pois nem sempre a delimitação das palavras é feita apenas com espaços, a

delimitação pode ocorrer também com sinais de pontuação, por exemplo: pontos, vírgulas, aspas, hífens e outras marcações. Essas pontuações podem gerar ambiguidade no momento da criação dos tokens, pois podem delegar funções diferentes em uma frase. Sendo assim, o tokenizador utilizado deve estar preparado para receber sinais de pontuação e determinar quando um sinal faz parte do token ou quando é apenas um símbolo (INDURKHYA; DAMERAU, 2010). 

2.3.1.1.2 Segmentação de frase



Na maioria dos idiomas, sinais de pontuação delimitam o fim de uma frase, porém essa regra nem sempre é bem delimitada, fazendo com que o segmentador de frases possa ter um mal entendimento em quando uma frase foi finalizada ou não. A complexidade da execução desta etapa está totalmente dependente do idioma utilizado pelo segmentador (INDURKHYA; DAMERAU, 2010).

Sendo assim, tende a ser arbitrária a definição do que constitui uma frase, ficando em muitos casos a cargo do desenvolvedor que está implementando o sistema definir quais serão as regras a serem utilizadas para limitar uma frase. Entretanto, os sistemas que utilizam PLN, em sua grande maioria, utiliza um método que consiste em verificar espaços seguidos por uma palavra iniciada com letra maiúscula seguido até um ponto final, interrogação, exclamação entre outros pontos, para delimitar o início e fim de uma frase (INDURKHYA; DAMERAU, 2010).

2.3.1.2 Análise léxica



O processo de decompor um texto em palavras, frases e outros elementos significativos é também definido como análise léxica. Nesta análise é baseada ao nível de palavra, ou seja, o foco é no significado das palavras, frases, e outros elementos, como os símbolos. Em alguns momentos, a análise léxica é vagamente descrita como um processo de tokenização (THANAKI, 2017).



Na análise léxica, duas etapas são muito comuns: *lemming* e *stemming*:

- *Lemming*: nesta etapa são relacionadas diferentes ocorrências morfológicas de uma determinada palavra em uma única forma, ou seja, a forma mais básica de uma palavra, em outros termos, seu radical: *lema*. Exemplo: comido, comeu, comendo, são do mesmo *lema* comer (INDURKHYA; DAMERAU, 2010).
- *Stemming*: é uma etapa mais bruta, onde o final de uma palavra é removido com o objetivo de reduzi-la a forma mais básica para encontrar outras ocorrências dessa palavra em diferentes formatos ao longo do texto (STANFORD, 2008).

Ambas as etapas compartilham do mesmo objetivo que é reduzir a palavra para sua forma mais básica. Porém, no *lemming*, geralmente necessita de ferramentas adicionais que lidem somente com essa tarefa, onde requer mais processamento, fazendo com que esse processo seja mais viável em sistemas mais robustos. No caso de aplicações mais simples, é mais aconselhável a utilização do *stemming* por ser mais rápido, porém o seu uso pode causar perda de informação dependendo do que for removido de uma palavra, visto que é um método mais bruto (BAASCH, 2021).

2.3.1.3 Análise sintática



Na análise sintática, é feita uma análise gramatical no texto, onde é feita sobre uma sequência de palavras fornecidas, de modo **cumum** sendo um frase, é gerado uma estrutura de acordo com a gramática escolhida, esta estrutura é utilizada a fim de atribuir um significado a frase ou palavra. As palavras fornecidas normalmente serão palavras processadas na etapa de tokenização e na etapa de análise léxica. Nesta etapa também é atribuído *tags*, ou seja, a palavra é atribuída a uma classe gramatical. As *tags* facilitam quando uma informação pertinente precisa ser extraída do texto (INDURKHYA; DAMERAU, 2010).

2.3.1.4 Análise semântica

Encontrar significado para o texto é o papel da análise semântica. Nesta etapa, o computador tem o poder de entender e interpretar frases, parágrafos

ou documentos inteiros, analisando sua gramática e identificando as relações entre as palavras de uma frase em um contexto específico. Sendo assim, o objetivo principal da análise semântica é extrair do texto o significado exato ou o significado do dicionário das palavras (GOYAL, 2021).

Na análise semântica, visto que já se procura compreender o significado das sentenças, ela pode ser considerada uma análise pragmática. Nesta etapa, pode-se também ser modelada para a realização de resumos automáticos, mineração de dados e tradução automática (BAASCH, 2021).

Conforme a análise for mais robusta e refinada, mais simples se torna a compreensão dos dados de entrada fornecidos pelo usuário, tornando assim mais eficiente a interação entre humano e máquina (INDURKHYA; DAMERAU, 2010).

2.3.2 Abordagem estatística

A abordagem estatística para o PLN utiliza de técnicas de aprendizado de máquina, onde para desenvolver um sistema capaz de processar com linguagem natural são utilizados conjuntos de dados com um enorme número de registros, esse conjunto de dados é fornecido a um algoritmo que busca padrões dentro os dados. Ao encontrar padrões, eles passam a fazer parte de um modelo que possui a capacidade de compreender os dados de entrada fornecidos para o processamento de linguagem natural (INDURKHYA; DAMERAU, 2010).

O domínio estatístico muitas vezes leva o PNL a ser descrito como Processamento Estatístico de Linguagem Natural, para que não haja confusão ao associá-lo aos métodos da abordagem clássica (BROWNIEE, 2017).

A popularidade da abordagem estatística tem aumentado nos últimos anos por não necessitar de conhecimentos tão especializados, pois não exige uma análise muito aprofundada, sendo necessário apenas possuir um volume de dados volumoso e uma classificação correta dos dados utilizados no treinamento (BAASCH, 2021).

X. REFERÊNCIAS BIBLIOGRÁFICAS

BAASCH. A. V. S, “**Aplicação de processamento de linguagem natural para análise de texto e indicação de notícias similares: uma ferramenta de apoio para a identificação de fake news**”, 2021

BARKER. D. “**12 open source tools for natural language processing**”, 2019. Disponível em: <https://opensource.com/article/19/3/natural-language-processing-tools> Acesso em 11 dez. 2021

BASILI, V. R. CALDIERA, G.; ROMBACH, H. D. “**Goal Question Metric Paradigm**”. In: MARCINIAK Encyclopedia of Software Engineering. [S.l.]: John Wiley & Sons, 1994.

BROWN R. B, “**Doing Your Dissertation in Business and Management: The Reality of Research and Writing**”, 2006. Sage Publications

BROWNIEE. J., “**What Is Natural Language Processing?**”, 2017. Disponível em: <https://machinelearningmastery.com/natural-language-processing/> Acesso em 16 abr. 2022

COHN, M. “**User stories applied for agile software development**”, 13. ed. Crawfordsville, Indiana. 2009. 263 p.

FEMMER, H. “**Reviewing Natural Language Requirements with Requirements Smells—A Research Proposal**”. Proceedings of IDoESE, 2013

FEMMER, H., FERNÁNDEZ, D.M., JUERGENS, E., KLOSE, M., ZIMMER, I., ZIMMER, J.: “**Rapid requirements checks with requirements smells: two case studies**”. In: Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering. pp.10–19. ACM, 2014

FERRARIO. A, NÄGELIN. M, **“The Art of Natural Language Processing: Classical, Modern and Contemporary Approaches to Text Document Classification”**, 2020

FRANCINO. Y. **“The essential guide to user story creation for agile leaders”**, TechBeacon, 2017. Disponível em:
<https://techbeacon.com/app-dev-testing/essential-guide-user-story-creation-agile-leaders> Acesso em 17 mar. 2022

FRANKLIN. B. **“Advice to a Young Tradesman”**, 1748. Disponível em:
<https://founders.archives.gov/documents/Franklin/01-03-02-0130> Acesso em 10 dez. 2021

GIL, A. C. **“Como elaborar projetos de pesquisa”**. São Paulo: Atlas, 2010. ISBN 5ª edição.

GLINZ, M. 2000, **“Improving the quality of requirements with scenarios”**. In: Proceedings of the World Congress on Software Quality (WCSQ), pp 55–60

GOYAL, C. **“Part 9: Step by Step Guide to Master NLP – Semantic Analysis”**, 2021, Disponível em:
<https://www.analyticsvidhya.com/blog/2021/06/part-9-step-by-step-guide-to-master-nlp-semantic-analysis/> Acesso em 14 abr. 2022

HECK, P. KLABBERS, M. VAN EEKELEN, M. C. J. D. **“A software product certification model,”** Software Quality Journal, vol. 18, no. 1, pp. 37–55, 2010.

HECK, P. ZAIDMAN A., **“A quality framework for agile requirements: a practitioner’s perspective”**, 2014

HEATH, F. **The trouble with user stories**. 2020, DZone. Disponível em: <https://dzone.com/articles/the-trouble-with-user-stories-1> . Acesso em 04 dez. 2021

IEEE Computer Society (2018), **“Systems and software engineering — Life cycle processes — Requirements engineering”**, Second Edition 2018-11 ISO/IEC/IEEE 29148

INDURKHYA, N.; DAMERAU, F. J. **“Handbook of natural language processing.”** [S.l.]: CRC Press, 2010. v. 2.

LUCASSEN, G. DALPIAZ, F. WERF, J. M. V. D, BRINKKEMPER. S. **“Improving agile requirements: the Quality User Story framework and tool”**, 2016

LUTKEVICH, B. **“Natural language processing (NLP)”**, 2021. Disponível em: <https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP> Acesso em 09 abr 2022.

NASCIMENTO, R. ARANHA, E. KULESZA. U, LUCENA. M, **“Requirements Smells como indicadores de má qualidade na especificação de requisitos: Um Mapeamento Sistemático da Literatura.”** 10.17771/PUCRio.wer.inf2018-40, 2018

O’LEARY Z. **“The essential guide to doing research”**, 2004. Sage.

PAI. A. **“What is Tokenization in NLP? Here’s all you need to know”**, 2020. Disponível em: <https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/> Acesso em 11 abr. 2022

RAHARJANA, I. K, HARRIS. F, JUSTITIA. A, **“Tool for generating behavior-driven development test-cases,”** J. Inf. Syst. Eng. Bus. Intell., vol. 6, no. 1, p. 27, Apr. 2020, doi: 10.20473/jisebi.6.1.27-36.

RAHARJANA, I. K, SIAHAAN. D e FATICHAH. C, **"User Stories and Natural Language Processing: A Systematic Literature Review,"** in IEEE Access, vol. 9, pp. 53811-53826, 2021, doi: 10.1109/ACCESS.2021.3070606.

RACKSPACE TECHNOLOGY, **"Dos chatbots à Alexa: a evolução do Processamento de Linguagem Natural"**, 2020. Disponível em: <https://www.rackspace.com/pt/solve/evolution-nlp> Acesso em 31 jan 2022.

REHKOPF. M. **"Histórias de usuários com exemplos e um template"**, 2020. Disponível em: <https://www.atlassian.com/br/agile/project-management/user-stories> . Acesso em 09 dez. 2021

ROBERTS, E. **"Natural Language Processing: History"**, 2004. Disponível em: https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/nlp/overview_history.html Acesso em 11 abr. 2022

SAUNDERS, M., LEWIS, P., & THORNHILL, A. **"Research Methods for Business Students"**, 2007, (6th ed.) London: Pearson.

SILVA, P. **"O que é automação e para que serve? Conversando com o CTO"**, 2019. Disponível em: <https://gobacklog.com/blog/o-que-e-automacao-e-para-que-serve/> . Acesso em 6 jan. 2022.

SOMMERVILLE, I. . **"Engenharia de software"**, 9. ed. Pearson. 2011. 529 p.

STANFORD. **"Stemming and lemmatization"**, 2008. Disponível em: <https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html> Acesso em: 14 abr. 2022

TAKE BLIP, **"Tudo sobre NLP: o que é processamento de linguagem natural e seus desafios na Inteligência Artificial"**, 2019. Disponível em: <https://www.take.net/blog/tecnologia/nlp-processamento-linguagem-natural/> . Acesso em 01 dez. 2021.

THANAKI. J. “Python Natural Language Processing”, 2017 Disponível em:
<https://www.oreilly.com/library/view/python-natural-language/9781787121423/f7f54f6d-8257-4904-9c8e-88d4ac491b94.xhtml> .

Acesso em 14 abr. 2022

THAYER, R. H. e DORFMAN, M.; **“Introduction to Tutorial Software Requirements Engineering”** in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p. 1-2.

WAKE, B. **“INVEST in Good Stories, and SMART Tasks”**, 2003. Disponível em: <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/> Acesso em 11 jan 2022.

WAUTELET. Y, HENG. S, KIV. S, KOLP. M, **“User-story driven development of multi-agent systems: A process fragment for agile methods,”** Comput. Lang., Syst. Struct., vol. 50, pp. 159–176, Dec. 2017, doi: 10.1016/j.cl.2017.06.007.