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

Alan V. C. Ensina
Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC)
Florianópolis, Santa Catarina, Brasil
alanvinicius.ce@gmail.com

Fabiane B. V. Benitti
Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC)
Florianópolis, Santa Catarina, Brasil
fabiane.benitti@ufsc.br

Mattheus da H. França
Departamento de Engenharia de Software (CEAVI)
Universidade do Estado de Santa Catarina (UDESC)
Ibirama, Santa Catarina, Brasil
mattheushora@gmail.com

RESUMO

Histórias de usuário são frases curtas e semiestruturadas capazes de ilustrar os requisitos de um software na perspectiva de um usuário. Mas escrever histórias de usuário sem qualidade pode trazer problemas no entendimento pela equipe de desenvolvimento. Sendo assim, é necessário encontrar uma forma de automatizar a avaliação dessas histórias afim de obter uma maior qualidade. Se tratando de automatização, o Processamento de Linguagem Natural (PLN) combina linguística, ciência da computação e inteligência artificial para estudar as interações entre sistemas computacionais e a linguagem natural humana. Porém devido ao alto número de tecnologias de PLN presente no mercado, qual seria a mais indicada para avaliar histórias de usuário? O presente trabalho tem como objetivo realizar um estudo comparativo para identificar qual ou quais tecnologias de PLN presente no mercado são mais adequadas para analisar critérios de qualidade em histórias de usuário. Para isso foi desenvolvido, além do estudo comparativo, uma API para avaliação de histórias de usuário que utilizam o template de Cohn e de Gherkin. Por fim, foi realizada uma avaliação das tecnologias processando através da API 80 histórias de usuário diferentes.

PALAVRAS-CHAVE

engenharia de software, histórias de usuário, processamento de linguagem natural, PLN

ABSTRACT

User stories are short, semi-structured sentences capable of illustrating software requirements from a user's perspective. But writing poor quality user stories can lead to problems in understanding by the development team. Therefore, it is necessary to find a way to automate the evaluation of these stories in order to obtain a higher quality. When it comes to automation, Natural Language Processing (NLP) combines linguistics, computer science and artificial intelligence to study interactions between computational systems and human natural language. However, due to the high number of NLP technologies on the market, which would be the most suitable for evaluating user stories? This paper aims to carry out a comparative study to identify which NLP technologies on the market are most suitable for analyzing quality criteria in user stories. For this purpose, in addition to the comparative study, an API was developed to evaluate user stories that use the Cohn and Gherkin templates. Finally, an evaluation of technologies was carried out by processing 80 different user stories through the API.

KEYWORDS

software engineering, user stories, natural language processing, NLP

1 Introdução

"Tempo é dinheiro" [1] 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.

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 [2].

Jason Brownlee [3] 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.

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 elicitação. Para Thayer [4], a elicitação 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 elicitação serão definidas as exigências, os recursos, os objetivos e as utilidades que o sistema deve cumprir.

A especificação de requisitos no desenvolvimento ágil pode ser feito por meio de histórias de usuário. 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 [5], "uma história de usuário descreve a funcionalidade que será valiosa para um usuário ou comprador de um sistema ou software".
Cohn [5] também 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. Dentro do contexto de histórias de usuário, seria possível avaliá-las utilizando soluções de PLN levando em consideração os critérios de qualidade? Se sim, qual a solução mais adequada para o PLN no contexto de histórias de usuário?
Neste sentido, o objetivo central desse trabalho é realizar um estudo comparativo entre pequenas soluções utilizando PLN para avaliar a qualidade de histórias de usuário nos idiomas português e inglês.

2 Estudo comparativo

Tendo em vista a falta de estudos relacionados a avaliação de histórias de usuário utilizando PLN, foi realizado um estudo comparativo entre algumas tecnologias. Para identificar as tecnologias a serem comparadas, são utilizadas algumas diretrizes de estudos sistemáticos.
O objetivo da revisão sistemática consiste em encontrar tecnologias que atendam a avaliação de histórias de usuário. Contudo, ressalta-se que essa pesquisa não realizará uma revisão sistemática por completa, mas sim utilizará de algumas diretrizes e práticas para auxiliar o estudo comparativo das tecnologias.

2.1 Questões de pesquisa

Foram definidas 3 questões de pesquisa que auxiliará no processo de definição de tecnologias de PLN.
QP1: Quais são as tecnologias presentes no mercado?
Tem como objetivo verificar quais as tecnologias mais citadas, empresa responsável pelo desenvolvimento da tecnologia e link para download.
QP2: Como a tecnologia é classificada dentro do contexto do PLN?
Tem como objetivo verificar qual a abordagem a tecnologia se aplica (clássica ou estatística) e quais etapas são utilizadas.
QP3: Quais são suas características?
Tem como objetivo verificar as características que a tecnologia possui, como linguagem de programação utilizada, idiomas disponíveis (inglês/português), empresas que utilizam, documentação e tipos de licença.

2.2 Processo de busca

Por se tratar de um processo de busca de tecnologias e não uma busca de artigos científicos, foi realizado uma pesquisa no Google na data de 16 de junho de 2022. Foi construída e utilizada a seguinte string de busca:

*("Tool" **OR** "Tools" **OR** "Ferramenta" **OR** "Ferramentas") **AND** ("PLN" **OR** "Processamento de Linguagem Natural" **OR** "NLP" **OR** "Natural Language Processing") **AND** ("free" **OR** "gratuita" **OR** "open source" **OR** "código aberto")*

A justificativa para a estrutura da string de busca se deve ao fato de possuir muitas soluções/ferramentas disponíveis que não são gratuitas, sendo assim, ficou limitado a busca de ferramentas open source ou que não haja custo no desenvolvimento da pesquisa.

2.3 Critérios de inclusão e exclusão

Para facilitar a pesquisa, foram definidos alguns filtros que ajudam a eliminar resultados irrelevantes e fora do escopo das questões, sendo assim, foram adotados alguns critérios de inclusão e exclusão.

Critérios de inclusão:
- A tecnologia processa textos nos idiomas português ou inglês
- A tecnologia não possui nenhum custo associado ao uso
- A tecnologia implementa totalmente ou parcialmente as etapas das abordagens clássica ou estatística

Critérios de exclusão:
- A tecnologia possui custo associado
- A tecnologia não processa textos nos idiomas português ou inglês

2.4 Execução

Como se trata de um levantamento de tecnologias presentes no mercado, a quantidade de dados retornados na busca é muito grande, com mais de 57.900.000 resultados, sendo assim, o levantamento foi feito considerando os 30 primeiros sites retornados, ou seja, os sites com maior relevância segundo o Google.
Após pesquisa, foram aplicados os 3 critérios de inclusão e descartadas as tecnologias que não se adequavam aos critérios preestabelecidos. Segue abaixo a Figura 1 exemplificando o método de pesquisa e aplicação dos critérios:

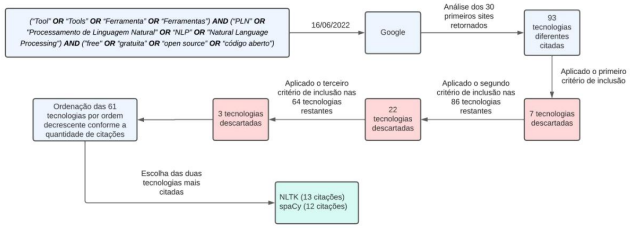


Figura 1: Processo de execução da pesquisa e análise das tecnologias

2.5 Resultados

A filtragem das tecnologias, de acordo com os critérios de inclusão e exclusão, aconteceu de acordo com três etapas conforme mostrado na Figura 1. Com relação a **QP1**, foram encontradas 93 tecnologias diferentes nos 30 sites analisados. Com relação a **QP2** e **QP3**, tendo em vista o alto número de tecnologias encontradas, foram analisadas as 10 tecnologias mais citadas, onde as 10 tecnologias foram classificadas com abordagem clássica e das principais características encontradas são que a maioria delas utilizam Python e Java em seu desenvolvimento.
Das 93 tecnologias encontradas, 61 se enquadram dentro dos critérios preestabelecidos. Como critério de desempate, as tecnologias foram ordenadas e contadas a quantidade de citações

dentro dos 30 sites analisados, e por fim, as duas tecnologias mais citadas foram selecionadas: NLTK e spaCy.
Após análise das tecnologias encontradas, foi decidido dar contiuidade a pesquisa com a implementação de dois protótipos, sendo um utilizando NLTK e o outro utilizando spaCy.
A escolha dessas tecnologias deve-se ao fato de serem as tecnologias mais consolidadas no mercado de PLN e por atenderem todos os requisitos buscados nas questões da pesquisa.

3 Desenvolvimento

Neste trabalho foi desenvolvido uma API capaz de avaliar histórias de usuário. A API foi desenvolvida utilizando as duas tecnologias que mais se destacaram no estudo comparativo realizado anteriormente: NLTK e spaCy.
Também foi utilizado o Swagger UI que é um framework open source e gratuito que permite visualizar e interagir com a API desenvolvida.
Lucassen [6] em seu estudo expõe 13 critérios de qualidade para avaliação de histórias de usuário divididos em 3 grupos: sintáticas, semânticas e pragmáticas. Em ambos os protótipos desenvolvidos, foram avaliados apenas os três critérios correspondentes ao grupo de critérios de qualidades sintáticas, ou seja:
- Bem-formada: a história de usuário possui apenas uma funcionalidade com um propósito
- Atômica: a história de usuário representa um requisito para exatamente um recurso
- Mínima: a história de usuário contém nada mais que um ator, uma ação e uma finalidade que pode ser opcional.

3.1 Pré-processamento

Antes de qualquer processamento de texto, o texto deve ser tratado afim de identificar erros que prejudiquem as análises. Tendo em vista que a análise será feita apenas sintaticamente e não semanticamente, o texto a ser processado deverá ser tratado antes da requisição ser enviada para a API.
Tendo como base o estudo realizado por Lucassen [6], para que as histórias e cenários sejam processados, é necessário seguir um padrão de palavras chave estruturando as frases antes do seu processamento, pois a mesma será subdividida em sentenças pela qual será possível identificar o ator/pré-condição, ação e a finalidade. Sendo assim, segue na Tabela 1 em negrito as palavras chaves que as histórias e cenários devem conter para que seja possível segmentar as sentenças que serão processadas.

Exemplo de história de usuário no template de Cohn [5] em português	"**Eu como** vendedor **gostaria** de cadastrar meus produtos **para que** eu possa listá-los posteriormente"
Exemplo de história de usuário no template de Cohn [5] em inglês	"**I as** a seller I **would like** to register my products **so** I can list them later."
Exemplo de história de usuário orientado a cenário no template de Gherkin [7] em português	"**Dado** que o cliente deseja abrir uma conta, informou o CPF, informou o RG e informou o endereço, **quando** entrar com essas informações no cadastro, **então** uma nova conta deve ser criada."
Exemplo de história de usuário orientado a cenário no template de Gherkin [7] em inglês	"**Given** a customer wants to open an account, and the ID was informed, and the address was informed, **when** all those information was typed, **then** a new account must be created."

Tabela 1: Exemplos de templates com as palavras-chave em negrito

Esse tipo de tratamento é primordial para que seja possível validar os critérios de qualidade, pois determinados critérios levam em consideração a ordem das sentenças, ou seja, no exemplo acima, ao validar qual o ator da história de usuário, espera-se que ele seja encontrado na primeira sentença da história: "Eu como vendedor".

3.2 Normalização entre idiomas

Para que seja possível validar as histórias de usuário nos idiomas português e inglês, foi necessário normalizar as classes gramaticais entre os idiomas, pois o idioma português possui diversos tipos de pronomes (pronome pessoal, demonstrativos, interrogativos, possessivos, relativos e indefinidos), já em inglês não há essa distinção entre pronomes.
Portanto, todas as classes gramaticais que possuem mais que um tipo foram agrupadas da maneira mais abrangente possível, ou seja, uma tag gerada através de PLN que seja um pronome pessoal, será considerado apenas como um pronome. Por exemplo, ao processar a frase: "Eu como vendedor gostaria de cadastrar meus produtos para que eu possa listá-los posteriormente.", cada palavra da frase recebe uma tag:

História	NLTK	spaCy
Eu como vendedor gostaria de cadastrar meus produtos para que eu possa listá-los posteriormente.	"Eu --> PROPESS --> PRONOME", "como --> PREP --> PREPOSIÇÃO", "vendedor --> N --> SUBSTANTIVO", "gostaria --> V --> VERBO", "de --> PREP --> PREPOSIÇÃO", "cadastrar --> V --> VERBO", "meus --> PROADJ --> PRONOME", "produtos --> N --> SUBSTANTIVO", "para --> PREP --> PREPOSIÇÃO", "que --> PROSUB --> PRONOME", "eu --> PROPESS --> PRONOME", "possa --> V --> VERBO", "listá-los --> N --> SUBSTANTIVO", "posteriormente --> ADV --> ADVÉRBIO", ". --> . --> INVÁLIDO"	"Eu --> PRON --> PRONOME", "como --> ADP --> PREPOSIÇÃO", "vendedor --> NOUN --> SUBSTANTIVO", "gostaria --> VERB --> VERBO", "de --> SCONJ --> CONJUNÇÃO", "cadastrar --> VERB --> VERBO", "meus --> DET --> ARTIGO", "produtos --> NOUN --> SUBSTANTIVO", "para --> SCONJ --> CONJUNÇÃO", "que --> SCONJ --> CONJUNÇÃO", "eu --> PRON --> PRONOME", "possa --> VERB --> VERBO", "listá-los --> VERB --> VERBO", "posteriormente --> ADV --> ADVÉRBIO", ". --> PUNCT --> INVÁLIDO"

Tabela 2: Exemplo de tags extraídas após o processamento com NLTK e spaCy

Na tabela 2 é possível observar que ao processar a mesma história em duas tecnologias diferentes, algumas tags são classificadas de maneiras diferentes entre as tecnologias, pode-se observar que a palavra "Eu" é um pronome pessoal, em NLTK a palavra recebe a tag PROPESS, já em spaCy recebe PRON. Sendo assim, para facilitar a análise, ambas as tags foram resumidas apenas como PRONOME. A avaliação da criação do POS Tagging por tecnologia será avaliada e detalhada na seção 4, em Avaliação.

3.3 Fluxo de processamento e validação

Para que se tenha uma imparcialidade ao avaliar as tecnologias, ambas seguem o mesmo fluxo de processamento e validação.

Na Figura 2 é possível observar que independente da tecnologia a ser processada, ambas seguem o mesmo fluxo para que não aja nenhuma parcialidade ao processar uma história ou cenário.

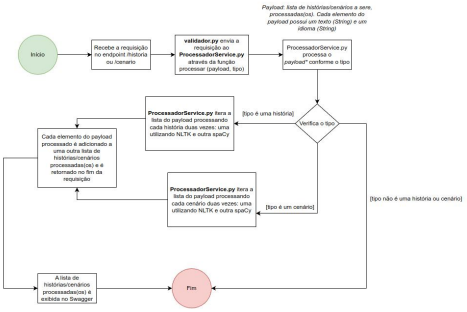


Figura 2: Fluxo de processamento geral

Na Figura 3 é possível observar o fluxo para o processamento de uma história de usuário que utiliza o template de Cohn [5]. O **ProcessadorService.py** chama a função *processarHistoria* duas vezes para a mesma história, uma passando o NLTK como tecnologia e outra o spaCy.
Na primeira etapa, o processador separa as sentenças utilizando a classe **UtilsService.py**, separando as sentenças conforme as palavras chave definidas na seção 3.1, em seguida é verificado qual a tecnologia veio por parâmetro. Caso seja NLTK, o **NLTKService.py** é chamado para processar e gerar os POS Taggings do texto e que serão utilizados para as validações. Caso seja spaCy, **SpacyService.py** é chamado para processar e gerar os POS Taggings.
Na segunda etapa, após processado o texto e gerado os POS Taggings, todas as validações a seguir são realizadas na classe **UtilsService.py**. A primeira validação feita é se a história é bem formada utilizando a função *verifica_C1_historia()*. Todas as regras para a validação dos critérios de qualidade estão descritas na seção a seguir. Em seguida é validado se a história é atômica utilizando a função *verifica_C2_historia()* e valida se a história é mínima utilizando a função *verifica_C3_historia()*.
Na terceira etapa, são extraídos o ator, a ação e a finalidade utilizando as funções: *extrair_ator(), extrair_ação* e *extrair_finalidade()*, também presentes no **UtilsService.py**. Essa extração não é utilizada para a validação dos critérios de qualidade, mas sim para a exibição na resposta do processamento.
Por fim, é verificado se há algum erro a ser exibido utilizando a função *verifica_erros_historia()* e finaliza o fluxo retornando a história processada e validada.

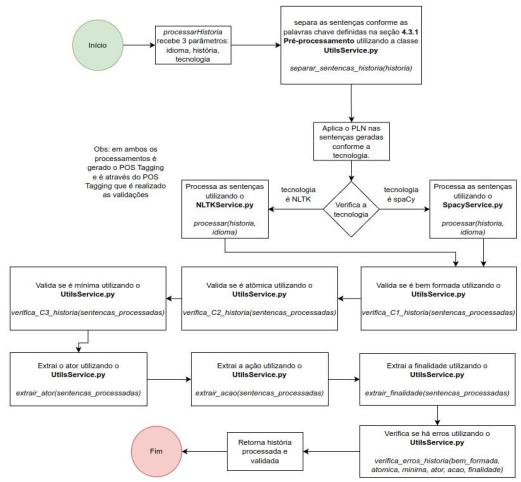


Figura 3: Fluxo de processamento para histórias de usuário que utilizam o template de Cohn [5]

Para o processamento de cenários que utilizam o template de Gherkin [7], pode-se observar na Figura 3 que o fluxo de processamento e validação é muito parecido com o da Figura 4 que utiliza o template de Cohn [5], porém utilizando funções específicas para processamento de cenários.

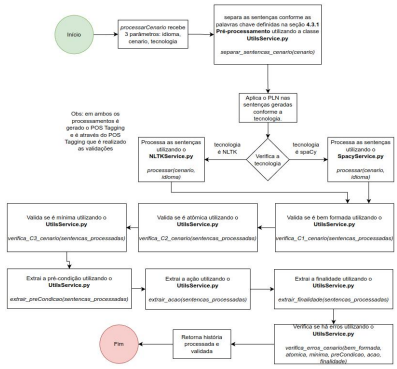


Figura 4: Fluxo de processamento para histórias de usuário que utilizam o template de Gherkin [7]

3.4 Regras de validação

Como foi dito anteriormente, os critérios de qualidade avaliados levam em consideração a parte sintática dos textos, sendo assim, para que seja possível aferir esses critérios, é levado em consideração as classes gramaticais de cada palavra em todas as sentenças processadas.
Para que seja possível descobrir qual a classe gramatical de cada palavra, será utilizado o POS-Tagging gerado por cada tecnologia em seu processamento.

3.4.1 Bem formada

Para validar o primeiro critério de qualidade, foi levado em consideração a estrutura definida por Lucassen [6], na qual uma história é bem formada quando há o seguinte formato: Quem realizará a tarefa + objetivo da tarefa + finalidade para a realização da tarefa (opcional).

Lucassen [6] em seu estudo informa que existem diversos templates possíveis para essa validação, porém, este trabalho segue o mesmo template que Lucassen [6] utilizou:

Sujeito + Adjetivo (opcional) + Verbo + Objeto indireto (opcional) + Objeto direto

Sendo assim, para validar se a história é bem formada, deve-se validar o sujeito (ator/pré-condição), o verbo (ação) e o objeto direto (finalidade).

3.4.1.1 Validação do ator

No caso de histórias de usuário que seguem o template de Cohn [5], o ator deverá ser identificado na primeira sentença e essa sentença deverá possuir as seguintes classes gramaticais:

substantivo + (pronome OU preposição OU artigo)

Caso a primeira sentença não possua um substantivo somado a um pronome, preposição ou artigo, a história não será bem formada pois haverá uma inconsistência ao encontrar o ator.

3.4.1.2 Validação da pré-condição

No caso de cenários, não será identificado o ator, mas sim uma pré-condição na primeira sentença. Para isso, é utilizado a mesma estrutura utilizada em histórias de usuário somados da presença da palavra Dado/Given:

Dado/Given + substantivo + (pronome OU preposição OU artigo)

Caso a primeira sentença não possua Dado/Given somados de um substantivo e um pronome, preposição ou artigo, o cenário não será bem formado pois haverá uma inconsistência ao encontrar a pré-condição.

3.4.1.3 Validação da ação

A validação da ação é o segundo critério a ser avaliado ao definir se uma história de usuário é bem formada. Para isso, ela segue duas estruturas diferentes: uma para o template de Cohn [5] e outra para a sintaxe de Gherkin [7].

Para o template de Cohn [5], a ação deverá ser encontrada na segunda sentença e é composta conforme a seguinte estrutura:

verbo + substantivo + pronome + (preposição OU advérbio)

No caso de cenários, a estrutura é semelhante, porém com alguns ajustes:

Pré-condição antes da ação + Quando/When + verbo + substantivo + (pronome OU preposição OU advérbio)

Neste caso, é validado se possui as palavras chaves da pré-condição (Dado/Given) antes da palavra chave da ação (Quando/When), se possui a palavra chave da ação (Quando/When), somados a um verbo, um substantivo e um pronome, preoposição ou advérbio. Caso não possua essa estrutura, a história não será bem formada pois haverá uma inconsistência ao encontrar a ação.

3.4.1.4 Validação da finalidade

No caso de histórias de usuários seguindo o template de Cohn [5], a finalidade é opcional, ou seja, poderá ou não estar presente na frase. Caso esteja presente, ela deverá ser encontrada na terceira sentença e seguirá a seguinte estrutura:

verbo + (pronome OU preposição OU substantivo OU advérbio)

Já no caso de cenários, a finalidade é obrigatória e deverá ser encontrada na sentença que se inicia com a palavra chave Então/Then.

Para validar a finalidade em cenários, a estrutura é a mesma do template de Cohn [5], porém é validado também a ordem das palavras chaves de todas as sentenças: Dado -> Quando -> Então:

Ordem correta das palavras chave + verbo + (pronome OU preposição OU substantivo OU advérbio)

Caso o cenário não possua essa estrutura, ele não será bem formado pois haverá uma inconsistência ao encontrar a finalidade.

3.4.2 Atômica

Segundo Lucassen [6], uma história de usuário é atômica quando há apenas um objetivo (ação) na tarefa. Sendo assim, para validar esse segundo critério de qualidade, primeiramente é identificado qual a sentença de ação da história/cenário.

Após identificado qual a sentença de ação, no caso de histórias de usuário que seguem o template de Cohn [5], é verificado se há alguma conjunção utilizando os conectivos e, ou, and e or. Ou seja, caso a história de usuário possua uma dessas conjunções será considerado como mais que uma ação, sendo assim viola a atomicidade da história de usuário.

No caso de cenários que seguem o template de Gherkin [7], a ação pode ser somada a condições, ou seja, a validação por meio de conectivos como e, ou, and e or não é válida. Sendo assim, para validar cenários, é verificado se a palavra-chave quando/when é utilizada mais que uma vez. Portanto, se no cenário for identificado mais que uma palavra-chave quando/when na sentença de ação, logo viola a atomicidade do cenário.

3.4.3 Mínima

O terceiro critério de qualidade avaliado é se a história de usuário é mínima. Uma história/cenário é mínima quando ela é bem formada (primeiro critério de qualidade) e não há informações extras, como comentários e notas adicionais [6].

Para validar o terceiro critério de qualidade, é verificado primeiramente se a história/cenário é bem formada e em seguida é verificado em todas as sentenças processadas se há algum caracter inválido, como por exemplo: *, [,], (,), {, }, _, :

Sendo assim, leva-se a entender que caso um desses caracteres esteja presente, significa que seja alguma nota adicional ao texto, portanto viola o critério de qualidade de minimalidade da história/cenário.

3.4.4 Exemplos de histórias válidas e inválidas

Para facilitar o entendimento de como uma história deve ser escrita, seja utilizando o template de Cohn [5] ou o template de Gherkin [7], segue abaixo a Tabela 3 com exemplos:

História	Bem formada	Atômica	Mínima
Eu como vendedor gostaria de cadastrar meus produtos para que eu possa listá-los posteriormente.	OK	OK	OK
Dado que o cliente deseja abrir uma conta, e informou o CPF, e informou o RG, e informou o endereço, quando entrar com essas informações no cadastro, então uma nova conta deve ser criada.	OK	OK	OK

Como vendedor gostaria de cadastrar.	Não é bem formada, pois na validação da ação não atendeu ao template	OK	Não é mínima se não é bem formada
Eu como vendedor gostaria de cadastrar meus produtos e listar na mesma tela.	OK	Não é atômica pois há mais que uma ação sendo realizada	OK
Eu como advogado gostaria de listar meus processos para que eu possa verificar quais estão em andamento. Nota: Não exibir os finalizados.	OK	OK	Não é mínima pois há uma informação adicional no final da história
Dado o corretor deseja listar, quando ele clicar então uma lista deve ser exibida.	Não é bem formada, pois na validação da ação não atendeu ao template	OK	Não é mínima se não é bem formada
Dado que uma pessoa deseja se cadastrar no site, quando ela clicar no botão novo usuário e quando ela preencher seus dados, então uma nova tela será exibida para ela inserir seus dados de cadastro.	OK	Não é atômica pois há mais que uma ação sendo realizada	OK
Dado que um motorista de aplicativo deseja iniciar uma corrida, quando uma nova notificação de corrida aparecer na tela e o motorista clicar em aceitar, então a localização para coleta do passageiro será exibida no GPS do aplicativo. Obs*: Apenas logado	OK	OK	Não é mínima pois há uma informação adicional no final da história

Tabela 3: Exemplos de histórias válidas e inválidas

4 Avaliação

Após desenvolvido o protótipo de avaliação de histórias de usuário, fez-se necessário avaliar ambas as tecnologias para aferir qual a mais indicada para essa avaliação. Sendo assim, foi utilizado a abordagem do GQM (Goal-Question-Metric) cujo resultado de sua aplicação especifica um sistema de medição visando um conjunto de questões e um conjunto de regras para interpretar os dados da medição [8].

O modelo GQM é uma estrutura hierárquica que pode ser definida de cima para baixo, através de uma meta define-se as perguntas que são respondidas através de métricas [8]. Pode-se observar melhor através da Figura 4, onde através da Meta 1 (Goal 1), são definidas três questões, que são respondidas através de 5 métricas.

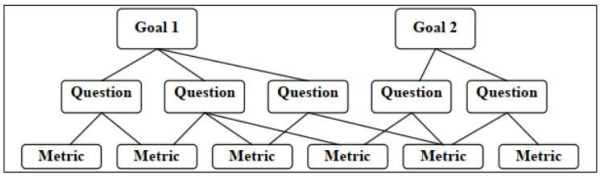


Figura 4: Exemplo de estrutura do GQM [8]

A seguir, é apresentado como foi realizado o planejamento para a avaliação, a execução e os resultados obtidos, e por fim, uma discussão a respeito dos resultados.

4.1 Planejamento

Para avaliar o protótipo e identificar qual tecnologia é a mais indicada para a avaliação de histórias de usuários, foram definidos três objetivos a serem analisados:

G1 - Analisar o protótipo com o propósito de avaliar o desempenho das tecnologias no contexto de tempo de processamento.

G2 - Analisar o protótipo com o propósito de avaliar a assertividade das tecnologias no contexto de processamento do POS Tagging.

G3 - Analisar o protótipo com o propósito de avaliar a eficácia das tecnologias no contexto de avaliação dos critérios de qualidade.

Para cada meta definida, foram associadas perguntas e a cada pergunta suas métricas, conforme pode-se observar nas tabelas 4, 5 e 6.

G1 - Analisar o protótipo com o propósito de avaliar o desempenho das tecnologias no contexto de tempo de processamento.
Q1.1 - Qual tecnologia tem o processamento mais rápido para o idioma Português (BR)?
M1.1.1 - Tempo total de processamento de histórias/cenários apenas em português para cada tecnologia
Indicador - A tecnologia que possuir o menor tempo de processamento para o idioma Português (BR)
Q1.2 - Qual tecnologia tem o processamento mais rápido para o idioma inglês?
M1.2.1 - Tempo total de processamento de histórias/cenários apenas em inglês para cada tecnologia
Indicador - A tecnologia que possuir o menor tempo de processamento para o idioma inglês

Tabela 4: GQM planejado para avaliar o tempo de processamento

G2 - Analisar o protótipo com o propósito de avaliar a corretude das tecnologias no contexto de processamento do POS Tagging.
Q2.1 - Qual tecnologia possui uma corretude maior ao definir o POS Tagging para o idioma Português (BR)?
M2.1.1 - Contagem de erros por tecnologia no processamento em português (BR)
Indicador - A tecnologia que possuir o menor número de erros
Q2.2 - Qual tecnologia possui uma corretude maior ao definir o POS Tagging para o idioma inglês?
M2.2.1 - Contagem de erros por tecnologia no processamento em inglês
Indicador - A tecnologia que possuir o menor número de erros

Tabela 5: GQM planejado para avaliar a assertividade do processamento do POS Tagging

G3 - Analisar o protótipo com o propósito de avaliar a eficácia das tecnologias no contexto de avaliação dos critérios de qualidade.
Q3.1 - Qual tecnologia possui maior eficácia ao avaliar o primeiro critério de qualidade: Bem formada
M3.1.1 - Contagem de histórias processadas com o primeiro critério válido
Indicador - A tecnologia que possuir a maior quantidade de acertos no processamento
Q3.2 - Qual tecnologia possui maior eficácia ao avaliar o segundo critério de qualidade: Atômica
M3.2.1 - Contagem de histórias processadas com o segundo critério válido
Indicador - A tecnologia que possuir a maior quantidade de acertos no processamento

Q3.3 - Qual tecnologia possui maior eficácia ao avaliar o terceiro critério de qualidade: Mínima
M3.3.1 - Contagem de histórias processadas com o terceiro critério válido
Indicador - A tecnologia que possuir a maior quantidade de acertos no processamento

Tabela 6: GQM planejado para avaliar a eficácia de avaliação dos critérios de qualidade

4.2 Execução

A avaliação do protótipo foi realizada no dia 05 de novembro de 2022 localmente utilizando um computador. As especificações do computador estão contidas na Tabela 7.

Processador	AMD Ryzen 7 3800X Cache 32MB 3.9GHz (4.5GHz Max Turbo)
Placa-mãe	Asus TUF B450M-Plus Gaming
Memória	XPG Spectrix D80, RGB, 2x16GB, 3200MHz, DDR4
Armazenamento	SSD Adata Falcon, 512GB, M.2
Placa de vídeo	Aorus AMD Radeon RX 5700 XT, 8GB, GDDR6
Alimentação	Fonte Corsair 750W 80 Plus Bronze

Tabela 7: Especificações do computador utilizado nos testes

Para a realização do teste foram criados 2 arquivos JSON, cada um deles contendo 40 histórias de usuário utilizando o template de Cohn [5] e 40 histórias de usuário utilizando o template de Gherkin [7]. Dentre as 40 histórias de cada template, 20 estão no idioma português (BR) e 20 em inglês. As 80 histórias criadas tiveram como referência um documento disponibilizado pelo Cohn com mais de 200 histórias escritas pelo próprio[9].
Foram realizadas duas requisições na API: uma utilizando o template de Cohn [5] com 40 histórias, e uma outra requisição utilizando template de Gherkin [7] com 40 histórias. E através dessas duas requisições foram gerados dois arquivos JSON utilizados na avaliação a seguir.

4.3 Resultados

A partir dos dados coletados na resposta das requisições feitas à API, foi possível responder a todas as perguntas relativas aos objetivos G1, G2 e G3. Das 80 histórias enviadas para o processamento, 12 histórias não teve processamento pois não atenderam aos templates de palavras-chave definidos definidos na seção 3.1.

Exemplo de história descartada	Como deveria estar
Sou vendedor e quero cadastrar meus produtos.	**Eu como** vendedor **gostaria** de cadastrar meus produtos.
Quando o cliente desejar abrir uma conta, quando entrar com as informações no cadastro, então uma nova conta deve ser criada.	**Dado que** o cliente deseja abrir uma conta, **quando** ele entrar com as informações no cadastro, **então** uma nova conta deve ser criada.

Tabela 8: Exemplos de descarte

A Tabela 8 apresenta duas das 12 histórias descartadas, onde pode-se observar a falta das palavras-chaves em negrito. Caso as histórias sejam enviadas sem as palavras-chave a API é incapaz de processar o texto e validar os critérios de qualidade, pois elas são responsáveis para segmentar as senteças que identificam o ator, a ação e a finalidade.
Das 12 histórias que não foram processadas, 6 foram identificadas na requisição ao endpoint de validação de histórias utilizando o template de Cohn [5] e 6 foram identificadas no endpoint de validação de histórias utilizando o template de Gherkin [7]. Sendo assim, a análise foi realizada considerando as 68 histórias que houveram processamento de linguagem natural. As 12 histórias não processadas foram descartadas da análise para que os resultados não fossem afetados.

4.3.1 Resultados referente ao G1 - Tempo de processamento

O primeiro objetivo consiste em analisar o protótipo com o propósito de avaliar o desempenho das tecnologias no contexto de tempo de processamento. Para isso, foram definidas duas questões a serem respondidas:
Q1.1) Qual tecnologia tem o processamento mais rápido para o idioma Português (BR)?
Q1.2) Qual tecnologia tem o processamento mais rápido para o idioma inglês?
Para responder ambas as questões foram levantados o tempo de processamento de cada uma das 68 histórias de usuário. Após feito o levantamento, foi possível calcular todas as métricas definidas para cada uma das questões.

M1.1.1 - Tempo total de processamento (ptbr)	
NLTK	spaCy
20,79037	22,40103
NLTK foi mais rápido que o spaCy em	7,19%

Tabela 9: Tempo total de processamento em português

Na Tabela 9 é possível observar que ao processar as 68 histórias de usuário no idioma português, o NLTK teve um tempo inferior que o do spaCy, sendo 7,19% mais rápido que o spaCy. Já na Tabela 10, ao processar as 68 histórias de usuário no idioma inglês, o NLTK também teve o tempo de processamento inferior ao do spaCy, neste caso, o NLTK foi 41,84% mais rápido que o spaCy.

M1.2.1 - Tempo total de processamento (inglês)	
NLTK	spaCy
19,36184	33,28852
NLTK foi mais rápido que o spaCy em	41,84%

Tabela 10: Tempo total de processamento em inglês

4.3.2 Resultados referente ao G2 - Corretude de processamento

O segundo objetivo consiste em analisar o protótipo com o propósito de analisar a corretude no processamento do texto e geração do POS Tagging. Para isso, foram definidas duas questões a serem respondidas:
Q2.1) Qual tecnologia possui uma corretude maior ao definir o POS Tagging para o idioma Português (BR)?
Q2.2) Qual tecnologia possui uma corretude maior ao definir o POS Tagging para o idioma inglês?
Para responder ambas as questões, foram definidas duas métricas:
M2.1.1 - Contagem de erros por tecnologia no processamento em português (BR)
M2.2.1 - Contagem de erros por tecnologia no processamento em inglês
Para realizar a contagem desses erros, foi analisado manualmente cada história de usuário processada e identificado cada erro por tecnologia, no final, foi contabilizado o total de erros para cada tecnologia e para cada idioma processado.

Na Tabela 11, é possível observar um exemplo de verificação em uma história de usuário que utiliza o template de Cohn [5]. Na tabela é identificado o POS Tagging gerado para o processamento utilizando NLTK e spaCy. Em seguida, é sinalizado os erros por tecnologia. Para auxiliar a validação das classes gramaticais, foram utilizados dois dicionários online que além de identificar o significado de cada palavra, identifica as classes gramaticais que a palavra pode pertencer. Para o idioma Português (BR), foi utilizado o DICIO[10], já para o idioma inglês, foi utilizado o DeepL[11].

História	POS Tagging		Erros gramaticais no POS Tagging	
	NLTK	spaCy	NLTK	spaCy
Eu como vendedor gostaria de cadastrar meus produtos para que eu possa listá-los posteriormente.	"Eu --> PROPESS --> PRONOME", "como --> PREP --> PREPOSIÇÃO", "vendedor --> N --> SUBSTANTIVO", "gostaria --> V --> VERBO", "de --> PREP --> PREPOSIÇÃO", "cadastrar --> V --> VERBO", "meus --> PROADJ --> PRONOME", "produtos --> N --> SUBSTANTIVO", "para --> PREP --> PREPOSIÇÃO", "que --> PROSUB --> PRONOME", "eu --> PROPESS --> PRONOME", "possa --> V --> VERBO", "listá-los --> N --> SUBSTANTIVO", "posteriormente --> ADV --> ADVÉRBIO", ". --> . --> INVÁLIDO"	"Eu --> PRON --> PRONOME", "como --> ADP --> PREPOSIÇÃO", "vendedor --> NOUN --> SUBSTANTIVO", "gostaria --> VERB --> VERBO", "de --> SCONJ --> CONJUNÇÃO", "cadastrar --> VERB --> VERBO", "meus --> DET --> ARTIGO", "produtos --> NOUN --> SUBSTANTIVO", "para --> SCONJ --> CONJUNÇÃO", "que --> SCONJ --> CONJUNÇÃO", "eu --> PRON --> PRONOME", "possa --> VERB --> VERBO", "listá-los --> VERB --> VERBO", "posteriormente --> ADV --> ADVÉRBIO", ". --> PUNCT --> INVÁLIDO"	"listá-los --> N --> SUBSTANTIVO",	"de --> SCONJ --> CONJUNÇÃO", "meus --> DET --> ARTIGO", "para --> SCONJ --> CONJUNÇÃO",

Tabela 11: Exemplo de avaliação de erros gramaticais no POS Tagging

No exemplo acima, o NLTK processou a palavra "listá-los" como um substantivo, ao invés de um verbo, já o spaCy processou "de" como uma conjunção ao invés de uma preposição, "meus" como um artigo ao invés de pronome e "para" como uma conjunção ao invés de uma preposição. Sendo assim, nesse exemplo o NLTK teve um erro gramatical e o spaCy teve três erros gramaticais.
Após levantado todos os erros de processamento de POS Tagging, foi contabilizado os erros por tecnologia e por idioma.
A Tabela 12 possui a métrica utilizada para responder a questão Q2.1, na qual o processamento em português, o NLTK teve 12 erros de POS Tagging, já o spaCy teve 55, sendo assim, o NLTK teve 78,18% menos erros que o spaCy.

M2.1.1 - Contagem de erros de corretude no POS Tagging (ptbr)		
	NLTK	spaCy
Erros identificados	12	55
Porcentagem de comparação de erros entre NLTK e spaCy		78,18%

Tabela 12: Contagem de erros de corretude de POS Tagging em português

Já para o processamento em inglês o resultado foi inverso. Na Tabela 13, pode-se observar a métrica utilizada para responder a questão Q2.2, na qual o processamento em inglês, o NLTK teve 14 erros, já o spaCy apenas 1, sendo assim o spaCy teve 92,86% menos erros em relação ao NLTK.

M2.2.1 - Contagem de erros de corretude no POS Tagging (inglês)		
	NLTK	spaCy
Erros identificados	14	1
Porcentagem de comparação de erros entre NLTK e spaCy		92,86%

Tabela 13: Contagem de erros de corretude de POS Tagging em português

4.3.3 Resultados referente ao G3 - Eficácia na avaliação dos critérios de qualidade

O terceiro e último objetivo consiste em analisar o protótipo com o propósito de avaliar a eficácia das tecnologias na avaliação dos critérios de qualidade. Para isso, foram definidas três questões a serem respondidas:
Q3.1) Qual tecnologia possui maior eficácia ao avaliar o primeiro critério de qualidade: Bem formada
Q3.2) Qual tecnologia possui maior eficácia ao avaliar o segundo critério de qualidade: Atômica
Q3.3) Qual tecnologia possui maior eficácia ao avaliar o primeiro critério de qualidade: Mínima
Para que seja possível responder à essas três questões foram definidas três métricas:
M3.1.1 - Contagem de histórias processadas com o primeiro critério válido
M3.2.1 - Contagem de histórias processadas com o segundo critério válido
M3.3.1 - Contagem de histórias processadas com o terceiro critério válido
Para que os critérios de qualidade sejam considerados válidos, a história precisa ser processada e independentemente se houve erro de processamento no POS Tagging, a validação do critério de qualidade deve ser aferida dentro das regras definidas nas seção 3.4, onde se encontram todas as regras e templates de validação de critérios de qualidade.

História (ptbr)	Erros gramaticais no POS Tagging		Processamento dos critérios de qualidade					
			Bem formada		Atômica		Mínima	
	NLTK	spaCy	NLTK	spaCy	NLTK	spaCy	NLTK	spaCy
Eu como vendedor gostaria de cadastrar meus produtos para que eu possa listá-los posteriormente	"listá-los --> N --> SUBSTANTIVO",	"de --> SCONJ --> CONJUNÇÃO", "meus --> DET --> ARTIGO", "para --> SCONJ --> CONJUNÇÃO",	VÁLIDA	INVÁLIDA	VÁLIDA	VÁLIDA	VÁLIDA	INVÁLIDA

Tabela 14: Exemplo de avaliação da eficácia na avaliação dos critérios de qualidade onde erros gramaticais afetaram a avaliação

Na Tabela 14, há um exemplo de como uma história que utiliza o template de Cohn [5] teve seu processamento avaliado. No exemplo, ao processar a história, a API deu como válido os três critérios de qualidade quando o processamento foi realizado com NLTK, porém, quando foi realizado com o spaCy, apenas o segundo critério de qualidade (Atômica) foi dado como válido.
Isso deve-se ao fato de que ao processar utilizando spaCy, os erros gramaticais identificados na terceira coluna influenciaram na validação dos critérios de qualidade, pois ao avaliar o primeiro critério de qualidade, na validação da ação da história, o template de validação não foi atendido, pois a classe gramatical das palavras processadas não correspondia com o template esperado: verbo + substantivo + preposição ou advérbio ou pronome.

NLTK	*spaCy*
"gostaria --> V --> VERBO",	*"gostaria --> VERB --> VERBO",*

"de --> PREP --> PREPOSIÇÃO", "cadastrar --> V --> VERBO", "meus --> PROADJ --> PRONOME", "produtos --> N --> SUBSTANTIVO",	"de --> SCONJ --> CONJUNÇÃO", "cadastrar --> VERB --> VERBO", "meus --> DET --> ARTIGO", "produtos --> NOUN --> SUBSTANTIVO",

Tabela 15: Comparação do POS Tagging na sentença de ação

Na Tabela 15 é possível observar que para NLTK, a sentença da ação foi validada corretamente, ao contrário do spaCy, na qual os erros de processamento do POS Tagging influenciaram na validação do template da ação, ou seja, foi identificado apenas o verbo e o substantivo, porém não identificou uma preposição ou advérbio ou pronome.

Após levantado e contabilizado o processamento das 68 histórias de usuário, foi analisado cada história individualmente e manualmente para verificar se o processamento da validação dos critérios de qualidade estavam corretos. Através desse levantamento, foi possível coletar as métricas utilizadas para responder as três questões: **Q3.1, Q3.2 e Q3.3**.

Contagem de critérios validados com sucesso (ptbr)						
	M3.1.1 - Bem formada (validação)		M3.2.1 - Atômica (validação)		M3.3.1 - Mínima (validação)	
	NLTK	spaCy	NLTK	spaCy	NLTK	spaCy
	34	27	34	34	34	28
Acertos	100%	79,41%	100%	100%	100%	82,35%

Tabela 16: Contagem de critérios validados com sucesso em português

Com relação ao processamento em português, na Tabela 16 é possível observar que a Q3.1, na avaliação do primeiro critério de qualidade bem formada, o NLTK obteve 100% de acertos na validação, enquanto o spaCy obteve 79,41% dos acertos. Quanto ao Q3.2, na avaliação do segundo critério de qualidade atômica, o NLTK também obteve 100% de acertos, assim como o spaCy que também obteve 100%. Com relação ao Q3.3, na avaliação do terceiro critério de qualidade mínima, o NLTK obteve 100% de acertos, já o spaCy obteve 82,35%.

Contagem de critérios validados com sucesso (inglês)						
	M3.1.1 - Bem formada (validação)		M3.2.1 - Atômica (validação)		M3.3.1 - Mínima (validação)	
	NLTK	spaCy	NLTK	spaCy	NLTK	spaCy
	34	34	34	34	34	34
Acertos	100%	100%	100%	100%	100%	100%

Tabela 17: Contagem de critérios validados com sucesso em inglês

Com relação ao processamento em inglês, pode-se observar na Tabela 17 que ambas as tecnologias obtiveram 100% de acertos.

4.3.4 Discussão

Após a avaliação e levantamento de todos os dados, foi possível responder a todas as questões propostas. A meta do G1 foi encontrar qual tecnologia tem o processamento mais rápido para os idiomas português e inglês. Em ambos os idiomas o NLTK teve um processamento mais rápido que o spaCy, sendo 7,19% para o português e 41,84% em inglês.

Um dos fatores que podem ter influenciado no resultado dessa performance é a forma em que foi implementado a API para processar um texto com NLTK e com spaCy. Na implementação com o NLTK, o desenvolvedor precisa definir quais etapas deseja utilizar ao processar o texto.

```
57
58    # Fluxo de processamento de texto no NLTK
59    def processar(texto:str, idioma:str):
60        tokens_palavras = NLTKService.tokenizar(texto, idioma)
61        lemas = NLTKService.lematizar(tokens_palavras)
62        pre_tags = NLTKService.tagging(lemas, idioma)
63        return utils.unificar_tagset(pre_tags, Constantes.NLTK)
```

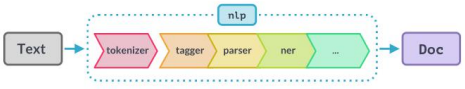
Figura 5: Etapas de processamento de texto utilizando NLTK

Conforme a Figura 5, ao processar um texto com o NLTK, foram definidas algumas etapas no processamento, como tokenizar, lematizar e por fim aplicar o POS Tagging. Já com o spaCy, pode-se observar na Figura 6, basta apenas definir o idioma e em seguida chamar uma função que faz todas as etapas.

```
7     def processar(texto, idioma):
8
9         if idioma == Constantes.EN:
10            nlp = spacy.load(Constantes.SPACY_EN)
11            doc = nlp(texto)
12            return utils.unificar_tagset(doc, Constantes.SPACY)
13
14        elif idioma == Constantes.PTBR:
15            nlp = spacy.load(Constantes.SPACY_PT)
16            doc = nlp(texto)
17            return utils.unificar_tagset(doc, Constantes.SPACY)
18
19        return None
20
```

Figura 6: Etapas de processamento de texto utilizando spaCy

Porém, ao analisar a documentação do spaCy, conforme Figura 7 mostra logo abaixo, observou-se que ao utilizar a função "nlp", várias informações são processadas, como por exemplo: tokens, parsers, NER - Named Entity Recognition (Reconhecimento de Entidade Nomeada), lematização, aplicação de labels. Essas informações extras podem influenciar no tempo de processamento da tecnologia, e, no contexto atual da API desenvolvida, essas informações extras são irrelevantes.



NAME	COMPONENT	CREATES	DESCRIPTION
tokenizer	Tokenizer	Doc	Segment text into tokens.
PROCESSING PIPELINE			
tagger	Tagger	Token.tag	Assign part-of-speech tags.
parser	DependencyParser	Token.head, Token.dep, Doc.sents, Doc.noun_chunks	Assign dependency labels.
ner	EntityRecognizer	Doc.ents, Token.ent_iob, Token.ent_type	Detect and label named entities.
lemmatizer	Lemmatizer	Token.lemma	Assign base forms.
textcat	TextCategorizer	Doc.cats	Assign document labels.
custom	custom components	Doc._.xxx, Token._.xxx, Span._.xxx	Assign custom attributes, methods or properties.

Figura 7: Pipeline de processamento presente na documentação do spaCy [12].

A meta do G2 consiste em analisar a corretude no processamento do texto e geração do POS Tagging para os idiomas português e inglês. Pôde-se observar que os resultados diferem com relação ao idioma. Ao processar o texto no idioma português, o NLTK teve

87,18% menos erros que o spaCy. Já ao processar em inglês, o spaCy teve 92,86% menos erros que o NLTK.
Essa diferença de resultados deve-se ao fato de que o NLTK não faz processamento de POS Tagging para português nativamente, ou seja, o NLTK apenas processa POS Taggings nos idiomas inglês e russo. Com isso, para que seja possível processar POS Taggins em português, foi necessário utilizar um conjunto de POS Taggers treinados para classificação gramatical em português.
Esse conjunto de POS Taggers foi encontrado no Github e incorporado a API para processamento de histórias de usuário utilizando NLTK no idioma português. O tagger utilizado foi o POS_tagger_brill.pkl que no teste realizado teve uma acurácia de 92.19% ao processar 30 mil palavras por segundo (INOUE, 2019).
Com relação a quantidade de erros identificados, no idioma português o NLTK teve 12 e o spaCy teve 55, já em inglês o NLTK teve 14 erros, contra apenas 1 no spaCy. Em português, se tratando de um conjunto de POS Taggers de terceiros utilizado no NLTK, esperava-se que a quantidade de erros fosse menor, pois o objetivo da criação desse POS Tagger foi preencher a lacuna que a tecnologia não atendia, mas teve uma quantidade de erros considerável comparado ao idioma inglês em que ambas as tecnologias processam nativamente.
A meta do G3 consiste em avaliar a eficácia das tecnologias nos critérios de qualidade para os idiomas português e inglês. Para o idioma português o critério de qualidade obteve 100% de eficácia para o NLTK e 79,41% para o spaCy. Já o segundo critério de qualidade em ambas as tecnologias tiveram 100% de acertos. Por fim, o terceiro critério de qualidade o NLTK teve 100% de acertos e o spaCy 82,35%. Para o idioma inglês, todos os critérios de qualidade de ambas as tecnologias tiveram 100% de acertos.
É possível observar que os resultados obtidos ao processar as histórias de usuário em português com spaCy não atingiram os 100% de acertos igual ao NLTK. Isso deve-se ao fato da quantidade elevada de erros de POS Tagging citadas no G2, pois a validação dos critérios de qualidade, principalmente do primeiro, é feita através das classes gramaticais que uma palavra possui. Outro ponto determinante na eficácia da validação do terceiro critério de qualidade é que este critério é dependente do primeiro, pois uma história não é mínima caso ela não seja bem formada [6].
Com base na análise dos resultados obtidos através do GQM, entende-se que ambas as tecnologias atendem ao objetivo proposto que foi a avaliação de critérios de qualidade em histórias de usuário. O NLTK teve maior destaque pois o tempo de processamento foi inferior ao do spaCy em português e inglês e a eficácia na validação dos critérios de usuário também foi superior. O spaCy se destacou apenas no processamento de texto em inglês, onde teve o menor número de erros ao gerar o POS Tagging responsável para a validação dos critérios de qualidade.

5 Considerações finais

Neste artigo foram analisadas algumas definições para histórias de usuário, critérios de qualidade e processamento de linguagem natural. Além desses conceitos, foi desenvolvido um estudo comparativo entre tecnologias de processamento de linguagem natural buscando identificar qual tecnologia é a mais indicada para a avaliação de critérios de qualidade em histórias de usuários.
No estudo comparativo foram definidas 3 questões de pesquisa para auxiliar na escolha das tecnologias. O objetivo das questões foi identificar quais tecnologias estão presentes no mercado, como ela é classificada dentro do contexto do PLN e quais suas características.
Foi realizado uma pesquisa no Google onde os 30 primeiros sites retornados foram utilizados no levantamento de tecnologias, onde 93 tecnologias foram identificadas. Foram definidos 3 critérios de inclusão na filtragem das tecnologias: processar texto em português ou inglês, não possuir custo associado, implementar totalmente ou parcialmente as etapas das abordagens clássica ou estatística.
Após aplicado os 3 critérios de inclusão, 64 tecnologias atendiam aos critérios. Por fim, a lista de 64 tecnologias foi ordenada em ordem decrescente na quantidade de citações, então, foi identificado que o NLTK obteve 13 citações e o spaCy 12.
Após identificado duas possíveis tecnologias de PLN, foi implementado um protótipo capaz de avaliar 3 critérios de qualidade sintáticas nas histórias de usuário: bem formada, atômica e mínima nos idiomas português e inglês.
Em seguida foi realizado uma avaliação comparativa utilizando o GQM. Através da avaliação foi possível observar que o NLTK teve um destaque maior que o spaCy quando o processamento foi feito em português. Quando o processamento foi executado para o idioma inglês, os resultados foram mais semelhantes.

5.1 Trabalhos futuros

Como sugestões de trabalhos futuros a serem desenvolvidos a partir dos resultados obtidos pode ser mencionado os seguintes pontos:
- Ampliar a quantidade de critérios de qualidade na avaliação, incluindo critérios de abordagem semântica.
- Inclusão de outras tecnologias de PLN no processamento das histórias de usuário, inclusive as de abordagem não supervisionadas, como o Google BERT.
- Utilizar histórias descartadas da avaliação para treinamento de modelos em tecnologias de abordagem não supervisionadas.

REFERÊNCIAS

[1] B. Franklin. In *"Advice to a Young Tradesman"*, 1748. https://founders.archives.gov/documents/Franklin/01-03-02-0130
[2] Rackspace Technology. In *"Dos chatbots à Alexa: a evolução do Processamento de Linguagem Natural"*, 2020. https://www.rackspace.com/pt/solve/evolution-nlp
[3] J. Brownlee. In *"What Is Natural Language Processing?"*, 2017. https://machinelearningmastery.com/natural-language-processing/
[4] R.H Tahyer and M. Dorfman. In *"Introduction to Tutorial Software Requirements Enginnering"* in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p. 1-2.
[5] M. Cohn. In *"User stories applied for agile software development"*, 13. ed. Crawfordsville, Indiana. 2009. 263 p.
[6] G. Lucassen, F. Dalpiaz, J. M. V.D. Werf and S. Brinkkemper. In *"Improving agile requirements: the Quality User Story framework and tool"*, 2016
[7] T. Hamilton, In *"Gherkin Language: Format, Syntax & Gherkin Test in Cucumber"*, 2022, https://www.guru99.com/gherkin-test-cucumber.html
[8] V. R. Basili, G. Caldiera, H. D. Rombach. In *"Goal Question Metric Paradigm"*. In: MARCINIAK Encyclopedia of Software Engineering. [S.l.]: John Wiley & Sons, 1994.
[9] M. Cohn. In *"User stories"*. 2020. https://www.mountaingoatsoftware.com/agile/user-stories
[10] Dicio, Dicionário online de Português. https://www.dicio.com.br/
[11] DeepL, Tradutor online. https://www.deepl.com/translator?utm_source=lingueecombr&utm_medium=linguee&utm_content=header_logo
[12] spaCy. In *"spaCy 101: Everything you need to know"*. *https://spacy.io/usage/spacy-101*