

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Inteligência Artificial e Aprendizado de Máquina

Gustavo Duran Ferreira

**UM FRAMEWORK WEB PARA RECOMENDAÇÃO DE TEXTOS UTILIZANDO
PROCESSAMENTO DE LINGUAGEM NATURAL**

São João da Boa Vista

2021

Gustavo Duran Ferreira

**UM FRAMEWORK WEB PARA RECOMENDAÇÃO DE TEXTOS UTILIZANDO
PROCESSAMENTO DE LINGUAGEM NATURAL**

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Inteligência Artificial e Aprendizado de Máquina como requisito parcial à obtenção do título de especialista.

São João da Boa Vista

2021

SUMÁRIO

1. Introdução	5
1.1. Objetivo Geral.....	5
1.2. Objetivos Específicos	5
2. Trabalhos Correlatos	6
2.1. Técnicas de processamento de linguagem natural aplicadas ao processo de mineração de textos: Resultados preliminares de um mapeamento sistemático	6
2.2. Classificação de documentos com processamento de linguagem natural ..	6
2.3. Comparação de ferramentas de processamento de linguagem natural para análise de sentimento em português: Um estudo de caso em avaliações online de hotéis	7
2.4. <i>BRAT: a Web-based Tool for NLP-Assisted Text Annotation</i>	7
2.5. <i>CNNs for NLP in the Browser: Client-Side Deployment and Visualization Opportunities</i>	7
2.6. NL2Type: Inferring JavaScript Function Types from Natural Language Information	8
3. Projeto do framework web para recomendação	8
3.1. Arquitetura Computacional	8
3.2 Etapas do processamento.....	9
3.2.1 Entrada.....	9
3.2.2 Normalização.....	10
3.2.3 Métricas	11
3.2.4 Armazenamento	13

3.2.5 Recomendação.....	15
4. Estudo de caso.....	17
4.1. Base de dados.....	17
4.2. Experimentos	18
4.2.1. Verificar similaridade com textos aleatórios.....	18
4.2.2. Verificar mudança de similaridade ao comparar recomendações	20
4.2.3. Verificar a similaridade de textos com contexto próximo.....	21
5. Análise de Resultados e Conclusões	23
REFERÊNCIAS	25

1. Introdução

O aumento da popularidade de internet, nos últimos anos, ocasionou um aumento na quantidade de informações disponíveis. Essa quantidade gera, por consequência, uma sobrecarga de informação em que se faz necessário uma análise de dados para encontrar as mais relevantes [1].

O Processamento de Linguagem Natural (PLN) tem como objetivo estudar a capacidade e as limitações dos computadores entenderem a linguagem humana. Sendo uma subárea da Inteligência artificial, existem diversos estudos no campo, visando a extração de informação [2].

Existem diversas técnicas de PLN para classificação de documentos, sendo que as principais *features* podem ser descobertas a partir da frequência das palavras, estrutura das palavras e estrutura da linguagem natural [3].

A motivação desta pesquisa surge ao tentar suprir a necessidade de entregar uma solução de classificação e recomendação generalista a base de dados formadas por conteúdo textual, utilizando técnicas de processamento de linguagem natural para contextualização dos dados e uso de métricas.

1.1. Objetivo Geral

O objetivo desta pesquisa é propor um *framework* web para classificação e recomendação de documentos, utilizando-se de técnicas de processamento de linguagem natural, visando apresentar à similaridade entre documentos construindo uma base contextualizada.

1.2. Objetivos Específicos

- Pesquisar sobre ferramentas de classificação e recomendação existentes;
- Estudar tecnologias que podem ser adotadas para a construção de ferramentas de classificação e recomendação, utilizando-se de PLN;
- Projetar a ferramenta de classificação e recomendação proposta;
- Experimentar a ferramenta proposta em um estudo de caso.

2. Trabalhos Correlatos

Essa seção tem o objetivo de apresentar os trabalhos correlatos, destacando a importância do uso do processamento de linguagem natural para análise de textos, utilizando técnicas disponível atualmente e as vantagens obtidas com essas metodologias para direcionar o desenvolvimento do framework no trabalho.

2.1. Técnicas de processamento de linguagem natural aplicadas ao processo de mineração de textos: Resultados preliminares de um mapeamento sistemático

A mineração de textos é a atividade que surgiu com o propósito de descobrir conhecimento em dados não estruturados (textuais). Este estudo teve como objetivo identificar e avaliar as técnicas de PLN disponíveis para realizar mineração em bases de dados textuais com o intuito de discutir sobre essas técnicas a partir das experiências publicadas neste contexto. Foram analisados 24 estudos aplicando 11 técnicas diferentes de PLN na mineração de textos, sendo que dentre todas essas técnicas, a ontologia se mostrou a mais recorrente e eficiente [4].

2.2. Classificação de documentos com processamento de linguagem natural

A classificação dos documentos de forma automática surge então como uma necessidade, visto um sistema automático ser capaz de classificar milhares de documentos em apenas alguns segundos. O maior desafio nesta área é obter os melhores resultados, maior eficiência computacional e maior capacidade de autoaprendizagem [3].

Um dos principais objetivos deste projeto é aplicar os classificadores implementados na classificação de literatura na área das proteínas. Este trabalho analisa e compara técnicas de classificação com processamento de linguagem natural, através da linguagem Python e com apoio da biblioteca para processamento de linguagem natural Natural Language Toolkit (NLTK). Foram avaliados em vários casos de estudo e os resultados obtidos nos diferentes estudos atingiram valores médios acima de 90% de precisão [3].

2.3. Comparação de ferramentas de processamento de linguagem natural para análise de sentimento em português: Um estudo de caso em avaliações online de hotéis

A nota de uma avaliação não traz toda a expressividade da avaliação do usuário, com isso encontra-se a necessidade de aplicar uma análise nos textos das avaliações online para obter melhores parâmetros para uma análise da Reputação Online deste Hotel [2].

Este trabalho aplica ferramentas de processamento de linguagem natural e ferramentas de Aprendizado de máquina para realizar uma análise de sentimento nas avaliações online de hotéis [2].

O estudo buscou por um website que disponibilizava avaliações online de hotéis e aplicou de métodos de web scrapping para a extração desses dados, unificando em uma base de dados com cerca de quatro mil e quinhentos registros, possibilitando a consulta dos dados para trabalhos futuros que visem a comparação de outras ferramentas e técnicas, ou até mesmo a utilização das mesmas ferramentas em outro domínio [2].

2.4. *BRAT: a Web-based Tool for NLP-Assisted Text Annotation*

Uma ferramenta baseada na web para anotação de texto suportada por PLN, com o objetivo de diminuir os esforços manuais e aumentar a produtividade de anotações. Foi discutido vários estudos de caso de projetos de anotação do mundo real usando versões de pré-lançamento de BRAT, mostrando uma diminuição de 15% no tempo total de anotação. BRAT está disponível sob uma licença de código aberto em: <http://brat.nlplab.org> [5].

2.5. *CNNs for NLP in the Browser: Client-Side Deployment and Visualization Opportunities*

Demonstra uma implementação de uma rede neural convolucional que realiza inferência feedforward completamente no navegador utilizando JavaScript. Este design é útil para aplicativos com requisitos de latência rigorosos ou baixa conectividade, pois essa implantação pode ser executada completamente no cliente, em uma ampla gama de dispositivos, sem fazer solicitações a servidor. Permite uma integração com o ecossistema JavaScript para visualização de informações, oferecendo

oportunidades de inspecionar visualmente as redes neurais e compreender melhor seu funcionamento interno [6].

2.6. NL2Type: Inferring JavaScript Function Types from Natural Language Information

Esta pesquisa apresenta uma abordagem baseada em aprendizagem para prever prováveis assinaturas de tipo de funções JavaScript, explorando informações de linguagem natural no código-fonte. Foi avaliado 162.673 arquivos de projetos do mundo real, prevendo tipos com uma precisão de 84,1% e um recall de 78,9% ao considerar apenas a sugestão mais alta, e com uma precisão de 95,5% e um recall de 89,6% ao considerar as 5 sugestões principais. O NL2Type supera as abordagens do JSNice e DeepTyper, outras duas ferramentas com o objetivo de previsão de tipos baseado em aprendizado de máquina [7].

3. Projeto do framework web para recomendação

O objetivo desse capítulo é apresentar o projeto conceitual do framework desenvolvido, visando destacar quais serão os passos para a implementação do uso de processamento de linguagem natural.

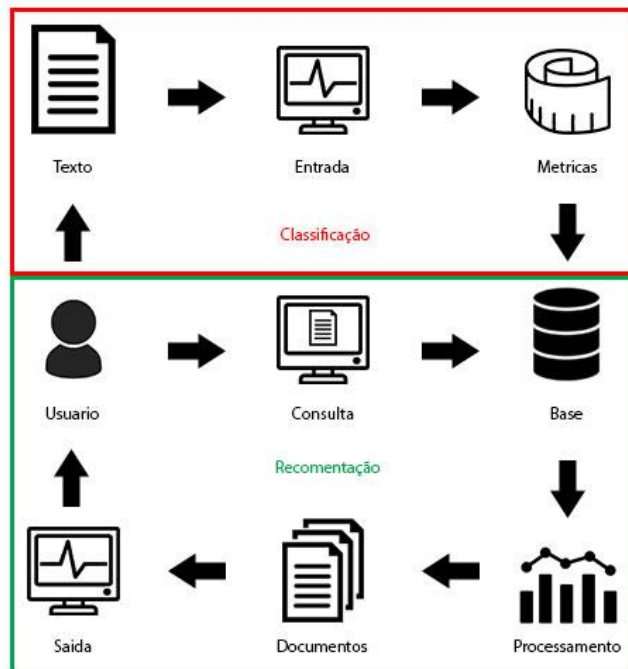
3.1. Arquitetura Computacional

Ao utilizar o framework, o usuário pode fazer uma entrada de um texto que será processado e convertido em métricas que serão armazenadas em uma base de dados. Esse processo de entrada pode ser utilizado para classificar os textos de forma que otimize o processo de recomendação posterior.

Com uma base de dados formada, o usuário pode fazer uma consulta dos textos armazenados e, com base nas métricas gravadas, receber uma saída de retorno com textos recomendados similares ao da consulta, utilizando processamento de linguagem natural.

A Figura 1 apresenta o fluxo da informação desde a entrada de um novo texto até as recomendações apresentadas.

Figura 1 - Projeto do framework



Fonte: Elaborada pelo Autor

3.2. Etapas do processamento

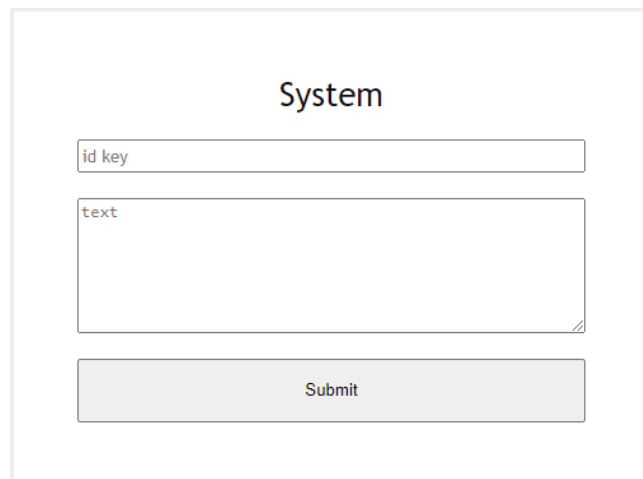
O objetivo dessa sessão é detalhar as etapas do processo para o uso do *framework*, visando destacar as conversões realizadas das informações até a obtenção do resultado de recomendação.

3.2.1. Entrada

A etapa de entrada é responsável por converter um texto em métricas que o representem na base de dados, com o uso de técnicas de processamento de linguagem natural. Essas métricas serão armazenadas para o uso no processo de recomendação.

Para isso, é utilizado uma interface web demonstrada na Figura 2 que aceita como entrada o texto e uma chave identificadora. A chave identificadora será utilizada como saída do processo de recomendação.

Figura 2 - Interface de Entrada



A interface de entrada do sistema, intitulada "System", contém um campo de texto rotulado "id key", um campo de texto rotulado "text" com uma seta no canto inferior direito, e um botão "Submit" cinza.

Fonte: Elaborada pelo Autor

Ao efetuar o envio do texto para o sistema, é iniciado o processo de normalização e geração de métricas através da linguagem Javascript. A normalização tem o objetivo de limpar o texto de forma que seja extraído somente o conteúdo relevante e as métricas tem o objetivo de representar a relevância desse conteúdo em relação a base.

3.2.2. Normalização

A normalização aplica uma série de técnicas de processamento de linguagem natural que converte o texto em termos. Um termo é uma palavra que será armazenada no contexto da base e cada um possuirá uma relevância.

Primeiro, é realizado o processo de tokenização, que retira todas as pontuações do texto e cria um vetor com todas as palavras do texto, gerando uma lista de termos. A segunda técnica aplicada é o processo de retirada das *stopwords*.

Uma *stopword* é uma palavra que aparecem com muita frequência e, portanto, possuem uma relevância baixa para o conjunto de dados. A lista de *stopwords*, em inglês, disponibilizada no link <https://gist.github.com/sebleier/554280> e apresentada na Figura 3, possui os 127 termos utilizados pela biblioteca NLTK, relevante na comunidade Python e foi adotada nesta pesquisa.

para o processamento são demonstradas na Figura 5. As repetições dos termos únicos são contabilizadas e utilizadas para calcular a métrica Term Frequency (TF) de cada termo do texto.

Figura 5 - Funções de processamento

```
const normalize = text => text
  .replace(/[^\w\s]/gi, " ")
  .replace(/(\n)/gm, ".")
  .replace(/(\t)/g, ".")
  .toLowerCase()
  .replace("length", "leng");
const tokenize = text => text.split(' ').filter(token => token !== "");
const cleaning = tokens => tokens.filter(token => stopwords.indexOf(token) < 0);
const stemming = tokens => tokens.map(token => stemmer(token));
const processing = text => stemming(cleaning(tokenize(normalize(text))));
const unique = tokens => {
  const words = [];
  for (token in count(tokens)) {
    words.push(token);
  }
  return words;
};
```

Fonte: Elaborada pelo Autor

O TF é a frequência do termo e representa a relevância de cada termo dentro do contexto do texto. O número é obtido ao dividir a quantidade de vezes que o termo aparece no texto pela quantidade de termos únicos no texto.

Outra métrica gerada é o Inverse Document Frequency (IDF), sendo que representa a importância do termo no contexto da base. O número é obtido ao dividir a quantidade de documentos da base pela quantidade de documentos que possuem o termo, as funções para cálculo das métricas são apresentadas na Figura 6.

Figura 6 - Funções de métricas

```
const count = tokens => {
  const words = [];
  tokens.forEach(token => words[token] = words[token] ? words[token] + 1 : 1);
  return words;
};
const tf = tokens => {
  const totalTokens = unique(tokens).length;
  const tokenTimes = count(tokens);
  const words = [];
  for (token in tokenTimes) {
    words[token] = tokenTimes[token] / totalTokens;
  }
  return words;
};
const idf = (tokens, db) => {
  const words = [];
  const totalTokens = db.map(tokens => count(processing(tokens)));
  unique(tokens)
    .forEach(token => {
      const aux = totalTokens
        .map(total => Number(total[token] > 0))
        .reduce((a, b) => a + b);
      return words[token] = db.length / aux;
    });
  return words;
};
```

Fonte: Elaborada pelo Autor

Ao realizar a multiplicação do TF com o IDF, é obtido a métrica TFIDF que representa a relevância daquele termo para o texto. Portanto, um array com o valor de TFIDF de cada termo da base para cada texto é uma opção de representar o documento utilizando uma métrica que expressa a relevância de seu conteúdo em relação a base.

A próxima etapa tem o objetivo de arquivar as informações de forma estruturadas para que possam ser utilizadas no processo de recomendação. A chave identificadora, o texto, os termos e o BoW são enviados para o processo de armazenamento.

3.2.4. Armazenamento

A etapa de armazenamento utiliza a linguagem PHP para receber os dados processados na etapa de classificação e se comunica um banco de dados MySQL para gravá-los. Foi adotado o PHP *Data Object* (PDO), extensão da linguagem PHP, pela possibilidade simplificada de alteração do banco de dados, podendo migrar o framework para outras tecnológicas como Oracle ou Firebase.

A tabela *sources* tem o objetivo de conter a representação de cada texto, portanto, será gravado a chave identificadora, o texto, os termos e o BoW. A tabela *tokens* tem o objetivo de conter as métricas de cada termo referente ao contexto da base, logo, será gravado o termo e o valor de IDF. As 2 tabelas podem ser respectivamente visualizadas nas Figuras 7 e 8.

Figura 7 - Tabela de *sources*

id	name	content	tokens	uniques
3	https://universe.l	Whether mistaken for a demon or god, many tales ha...	aatrox,aatrox,aatrox,aatrox,ε	aatrox,abil,advanc,also,alwai,ancient,a
4	https://universe.l	Noxian by birth, Shieda Kayn and others like him w...	abandon,abl,accept,acolyt,a	abandon,abl,accept,acolyt,ag,also,are:
5	https://universe.l	Regardless of what he would later become, Varus wa...	abandon,act,aliv,alon,amid,a	abandon,act,aliv,alon,amid,among,anc
6	https://universe.l	Tryndamere came into the world knowing only the ha...	abl,accept,affect,almost,amc	abl,accept,affect,almost,among,ancien
7	https://universe.l	Ashe hails from the northern Freljord, where bruta...	abandon,abil,accept,adopt,a	abandon,abil,accept,adopt,affect,allian
8	https://universe.l	Sejuani was the child of a Freljordan political m...	abandon,abl,absolut,absorb,	abandon,abl,absolut,absorb,across,affi
10	https://universe.l	In a time long forgotten, before the sands birthed...	abil,abl,abl,abyss,across,ag,	abil,abl,abyss,across,ag,allegi,alli,alwa
11	https://universe.l	One of the Notai, a nomadic tribe that long travel...	abil,accident,adventur,ago,a	abil,accident,adventur,ago,alon,alreadi
12	https://universe.l	Anivia is an ancient Freljordan demi-god who repr...	abl,accord,accord,across,ag	abl,accord,across,ag,almost,alwai,amp
13	https://universe.l	Orn's name was once spoken throughout the lands t...	accept,accomplish,accompli	accept,accomplish,admit,aftermath,ag,
14	https://universe.l	To some, he is the Thunder's Roar, the Greatstorm...	across,aftermath,agre,allow,	across,aftermath,agre,allow,alongsid,a
15	https://universe.l	Even as a child, Braum was much larger than other ...	abil,across,act,action,actual,	abil,across,act,action,actual,adopt,ado

Fonte: Elaborada pelo Autor

Figura 8 - Tabela de tokens

id	token	times
19675	aatrox	2
19676	abil	22
19677	advanc	10
19678	also	41
19679	alwai	62
19680	ancient	65
19681	annihil	8
19682	anoth	40

Fonte: Elaborada pelo Autor

Ao receber os dados, conforme apresentado na Figura 9, as informações referentes ao texto são gravadas como um novo registro na tabela sources. Em seguida, é realizado um processo que verifica todos os termos do novo texto inserido e grava um novo registro na tabela tokens para os termos novos com o valor de IDF sendo 1 ou incrementa 1 ao valor de IDF do termo existente. Dessa forma, os dados podem ser utilizados pelo processo de recomendação.

Figura 9 - Processo de armazenamento

```

$stmt = $pdo->prepare('
....INSERT INTO sources (name, content, tokens, uniques) VALUES (:name, :content, :tokens, :uniques)
');
$stmt->execute(array(
....:name' => $_POST['name'],
....:content' => $_POST['content'],
....:tokens' => $_POST['tokens'],
....:uniques' => $_POST['uniques']
));

$uniques = explode(",", $_POST['uniques']);
foreach ($uniques as $token) {
....$stmt = $pdo->prepare('SELECT times FROM tokens WHERE token = :token');
....$stmt->execute(array(
....:token' => $token
....));

....if ($stmt->rowCount() == 0) {
....$stmt = $pdo->prepare('INSERT INTO tokens (token, times) VALUES (:token, :times)');
....$stmt->execute(array(
....:token' => $token,
....:times' => 1
....));
....} else {
....$times = $stmt->fetchAll(PDO::FETCH_ASSOC);
....$stmt = $pdo->prepare('UPDATE tokens SET times = :times WHERE token = :token');
....$stmt->execute(array(
....:token' => $token,
....:times' => $times[0]['times'] + 1
....));
....}
}
}

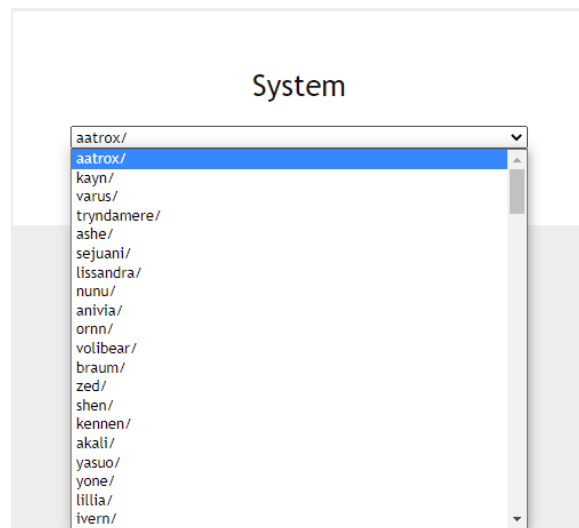
```

Fonte: Elaborada pelo Autor

3.2.5. Recomendação

A etapa de recomendação tem o objetivo de retornar os textos mais relevantes em relação a um determinado texto da base. Para isso, é utilizada outra interface que pode ser visualizada na Figura 10. A interface faz uma chamada para o PHP que é responsável pela comunicação com a base de dados e retorna a lista completa de textos cadastrados.

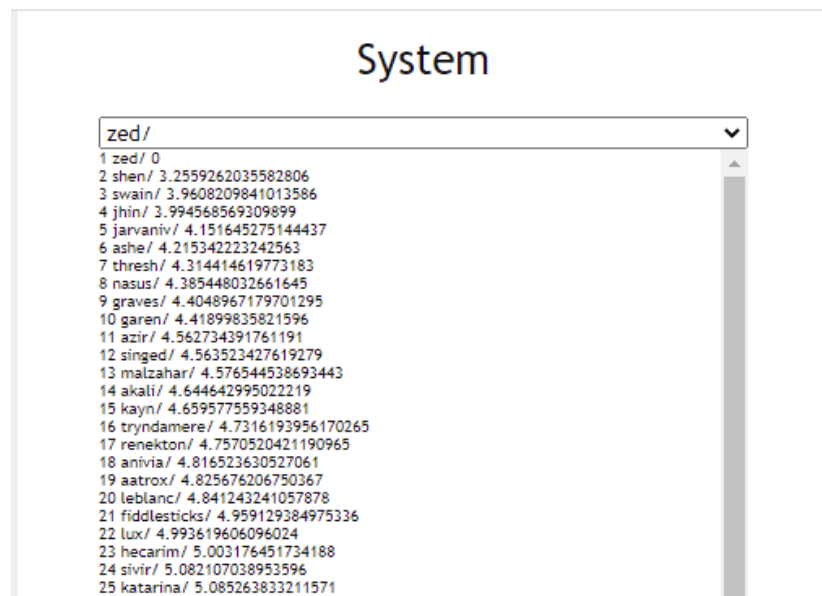
Figura 10 - Interface de consulta



Fonte: Elaborada pelo Autor

Ao selecionar um alvo, o Javascript realiza o calculado o vetor de TFIDF atualizado de cada texto, com os termos salvos na tabela sources para calcular o TF e o valores de IDF salvos na tabela tokens. É gerado uma lista de vetores da mesma quantidade de textos na base, cada vetor possui valores da possui da mesma quantidade de termos da base, conforme Figura 11.

Figura 12 - Interface de saída



Fonte: Elaborada pelo Autor

4. Estudo de caso

O objetivo dessa sessão é apresentar experimentos práticos da utilização do *framework* em uma base de dados, visando destacar os resultados obtidos ao recomendar textos com o processamento de linguagem natural.

4.1. Base de dados

Para realização do estudo de caso, foi definido que todos os textos possuiriam uma certa familiaridade por pertencerem ao mesmo escopo. Logo, foi selecionado uma origem única em que seria retirados todos os textos, sendo as biográficas de personagens do League of Legends.

O League of Legends é um jogo eletrônico, lançado em 2009 e desenvolvido pela Riot Games, em que duas equipes se enfrentam para destruir a base uma da outra. Atualmente, o jogo possui 154 personagens, cada um com uma história própria que pode ou não estar conectada com outros personagens. Todas as biografias estavam disponíveis no site oficial <https://universe.leagueoflegends.com>,

que descreve o universo de League of Legends e contém uma recomendação de outras biografias ao final de cada página com a história de um personagem.

4.2. Experimentos

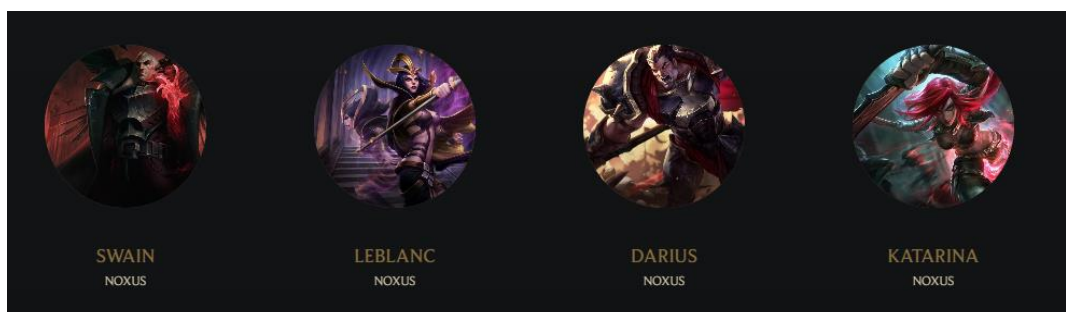
Após inserir todas as biografias do universo do jogo no *framework* desenvolvido, o objetivo dos experimentos é verificar a relação de distância entre os textos, validando a eficiência da recomendação com as histórias recomendadas pelo site oficial.

4.2.1. Verificar similaridade com textos aleatórios

O objetivo desse experimento é verificar as recomendações feitas pelo *framework* utilizando um personagem aleatório como alvo. O personagem selecionado foi o Swain, general do exército de Noxus.

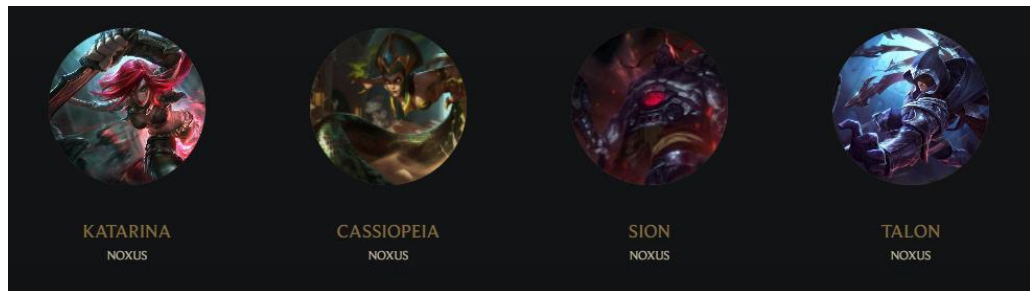
No final da biografia de Swain no site oficial, é recomendado as biografias de outros três personagens da região de Noxus, conforme Figura 13. Não existe uma garantia de que o site oficial indique a história de Swain ao acessar a biografia de qualquer um desses personagens, como pode ser visualizado na Figura 14 que contém a história de Katarina.

Figura 13 - Recomendação oficial de Swain



Fonte: Site oficial do League of Legends

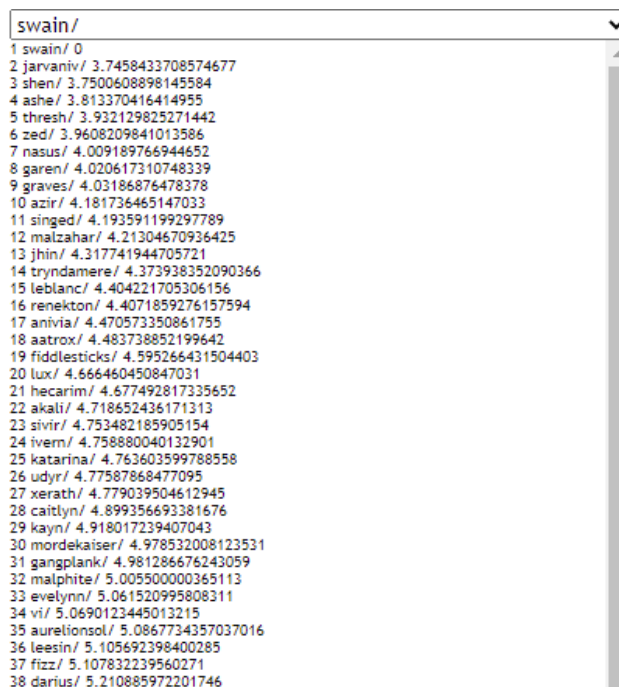
Figura 14 - Recomendação oficial de Katarina



Fonte: Site oficial do League of Legends

Ao utilizar o texto de Swain como alvo, o framework retorna os 154 textos em ordem de proximidade. A primeira posição pertence ao próprio texto alvo, sendo o texto mais próximo ao alvo o de Jarvan, com uma distância aproximada de 3,745. O primeiro personagem a parecer na lista igual ao indicado pelo site oficial é a LeBlanc, na posição 15 com uma distância aproximada de 4,404. Em seguida, conforme Figura 15, vemos a Katarina na posição 25 e Darius na posição 38, com distâncias aproximadas respectivas de 4,763 e 5,210.

Figura 15 - Recomendação do framework de Swain



Fonte: Elaborada pelo Autor

Pode-se verificar que ao utilizar as técnicas e processamento de linguagem natural, não foi obtido o mesmo resultado apresentado no site oficial. Com o resultado o *framework*, seria recomendado biografias de personagens que

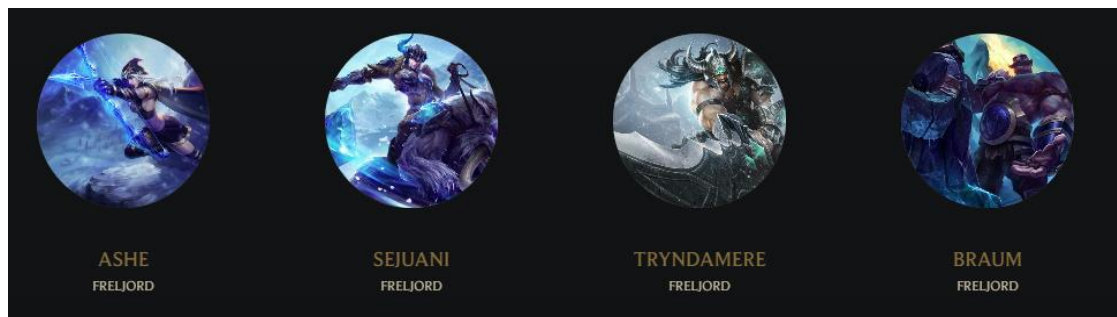
pertencem a outras regiões que podem ou não ter sua biografia conectada com a história de Swain.

4.2.2. Verificar mudança de similaridade ao comparar recomendações

O objetivo desse experimento é verificar a alteração de posição dos personagens na recomendação ao utilizar outro texto como alvo e comparar os resultados com o primeiro experimento.

O texto alvo selecionado foi a da Ashe, a arqueira de gelo pertencente a Freljord, que aparece na posição 4 como recomendação para o texto de Swain, com uma distância de aproximadamente 3,813. O site oficial recomenda histórias de outros três personagens de Freljord, conforme Figura 16.

Figura 16 - Recomendação oficial de Ashe



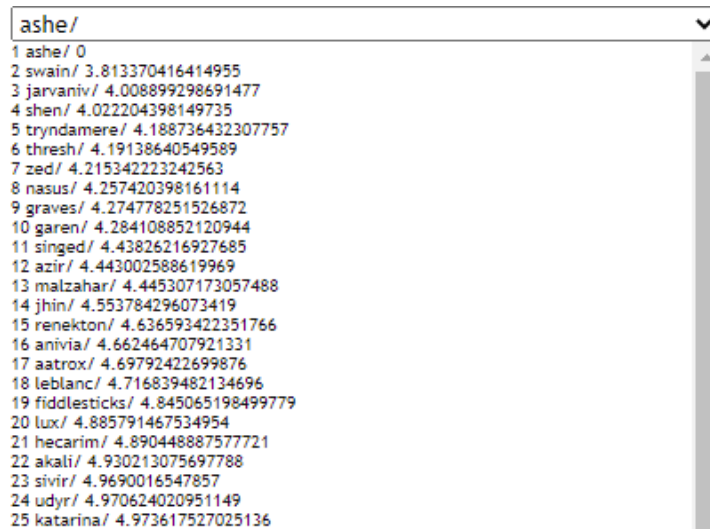
Fonte: Site oficial do League of Legends

Como no experimento anterior, a primeira posição pertence ao texto alvo. O framework recomenda os próximos textos como sendo o de Swain, Jarvan e Shen. Os personagens aparecem na mesma ordem do experimento anterior, antes possuindo as posições 1, 2 e 3, agora possuindo a posições 2, 3 e 4.

Agora na posição 4 possuímos o primeiro personagem igual ao site oficial, sendo o Tryndamere com uma distância aproximada de 4,188. Braum e Sejuani aparecem na posição 54 e 76, com distancias aproximadas de 6,122 e 6,944, respectivamente.

Apesar de algumas ordens permanecerem iguais entre a troca de alvo, pode-se verificar na Figura 17 que houve uma alteração de posição na recomendação dos personagens, pois Tryndamere ocupava a posição 14 no experimento anterior. Os personagens verificados no experimento 1 agora ocupam as posições 18, 25 e 38.

Figura 17 - Recomendação do framework de Ashe



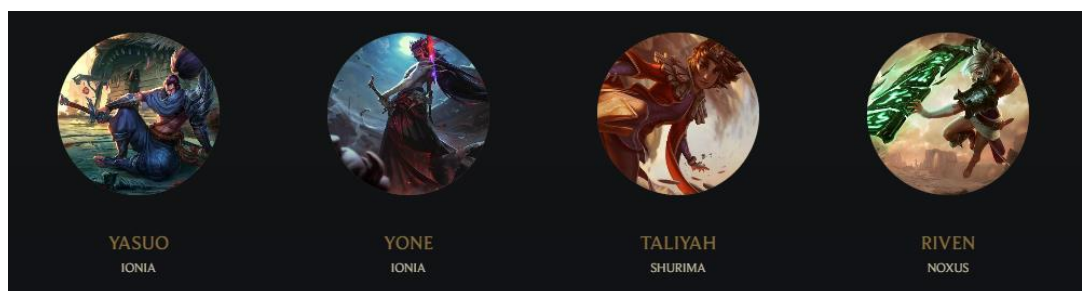
Fonte: Elaborada pelo Autor

4.2.3. Verificar a similaridade de textos com contexto próximo

O objetivo desse experimento foi a recomendação com base no conhecimento das biográficas dos personagens. Os experimentos anteriores não tiveram análises baseadas no contexto, foi analisado somente os retornos.

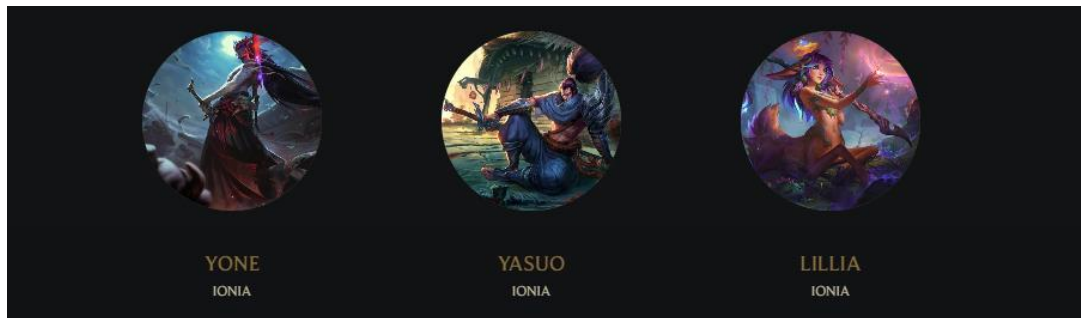
O texto alvo será o de Yasuo que possui uma proximidade contextual com a texto de seu irmão Yone. O site oficial apresenta a biografia de Yone como recomendação a biografia de Yasuo, conforme Figura 18, e o contrário também acontece apresentando Yasuo como recomendação de Yone, conforme Figura 19.

Figura 18 - Recomendação oficial de Yasuo



Fonte: Site oficial do League of Legends

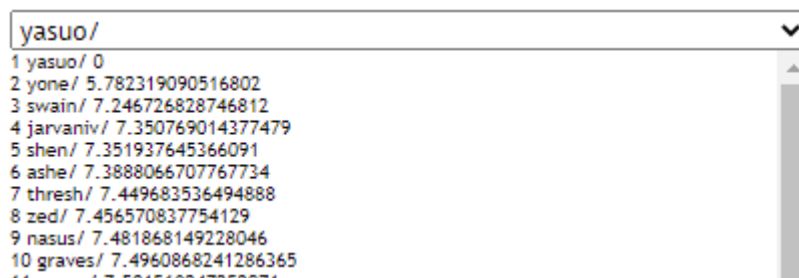
Figura 19 - Recomendação oficial de Yone



Fonte: Site oficial do League of Legends

Conforme verificado nos experimentos anteriores, a primeira posição da recomendação pertence ao texto alvo e em seguida é apresentado o texto de Yone, com uma distância aproximada de 5,782. A Figura 20 mostra que as próximas posições pertencem a Swain, Jarvan, Shen e Ashe. A recomendação de Yone demonstrada na Figura 21 segue a mesma regra de proximidade entre os irmãos, com alteração na ordem das recomendações seguintes.

Figura 20 - Recomendação do framework de Yasuo



Fonte: Elaborada pelo Autor

Figura 21 - Recomendação do framework de Yone



Fonte: Elaborada pelo Autor

Mesmo com uma distância maior entre a primeira e a segunda posição do que as apresentadas nos exemplos anteriores, um texto com contexto próximo como

sendo as biografias de irmãos, são apresentadas em sequência. Logo, o experimento mostra que as recomendações feitas pelo *framework* dependem da forma como os textos são escritos.

5. Análise de Resultados e Conclusões

Conforme demonstrado nos experimentos apresentados, o protótipo do *framework* web para recomendação é capaz de converter um texto em métricas de forma que possibilita uma comparação de relevância entre os documentos de uma base, utilizando técnicas de processamento de linguagem natural.

Os resultados obtidos nos experimentos demonstram que é possível recomendar textos que possuem conteúdos semelhantes. Portanto, o protótipo atende o requisito de ser uma ferramenta generalista que é capaz de recomendar textos de forma contextualizada, mas sofre de uma recomendação viciada com documentos que possuem um contexto genérico ou central em relação a base.

Visto isso, é possível implementar melhorias na extração de *features* para contextualizar de forma mais eficiente a entrada dos textos para que seja diminuído esse vício de recomendação da base. Também há possibilidade de melhoria no algoritmo de recomendação utilizando outras métricas que não foram testadas nesse trabalho.

REFERÊNCIAS

1. SILVA, T. M. S. **Extração de Informação para Busca Semântica na Web Baseada em Ontologias**. Mestrado em Engenharia Elétrica - Universidade Federal de Santa Catarina – UFSC, Florianópolis 2003.
2. PENCZKOSKI, R.; PENTEADO, R. J. **Comparação de Ferramentas de Processamento de Linguagem Natural para Análise de Sentimento em Português: Um Estudo de Caso em Avaliações Online de Hotéis**. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) – Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019.
3. SANTOS, C. M. **Classificação de Documentos com Processamento de Linguagem Natural**. Mestrado em Informática e Sistemas – Instituto Superior de Engenharia de Coimbra, 2015.
4. SANTOS, R. E. S. et al. **Técnicas de Processamento de Linguagem Natural Aplicadas ao Processo de Mineração de Textos: Resultados Preliminares de Um Mapeamento Sistemático**. Revista de Sistemas e Computação – Salvador, 2015.
5. STENETORP, P. et al. **BRAT: A Web-Based Tool for NLP-Assisted Text Annotation**. Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics – Avignon, 2012.
6. LIANG, Y. et al. **CNNs for NLP in the Browser: Client-Side Deployment and Visualization Opportunities**. Association for Computation Linguistics – New Orleans, 2018.
7. MALIK, R. S.; PATRA, J.; PRADEL, M. **NL2Type: Inferring JavaScript Function Types from Natural Language Information**. IEEE/ACM 41st International Conference on Software Engineering (ICSE) – Montreal, 2019.