PROCESSAMENTO DE LINGUAGEM NATURAL

AUTOR: FABRÍCIO GALENDE MARQUES DE CARVALHO

SUMÁRIO

1.INTRODUÇÃO

Nessa seção são elucidadas as principais motivações para o estudo do processamento de linguagem natural (PLN – Processamento e Linguagem Natural / NLP – do inglês, Natural Language Processing).

Além da motivação para o estudo, são discutidas algumas definições fundamentais, imprescindíveis para que o aluno possa entender e utilizar adequadamente documentações técnicas, são citadas algumas áreas de aplicação do PLN e, por fim, são abordadas as principais etapas envolvidas em uma linha de processamento de PLN.

1.1.MOTIVAÇÃO PARA O ESTUDO DE PLN

Um dos principais motivadores para o estudo do PLN é o **elevado volume de dados disponíveis** através dos sistemas informatizados. Este fato é decorrente da característica pervasiva e ubíqua dos sistemas computacionais que passaram a fazer parte dos mais diversos negócios e do dia a dia das pessoas.

Quando se fala em volume de dados, três características devem ser analisadas:

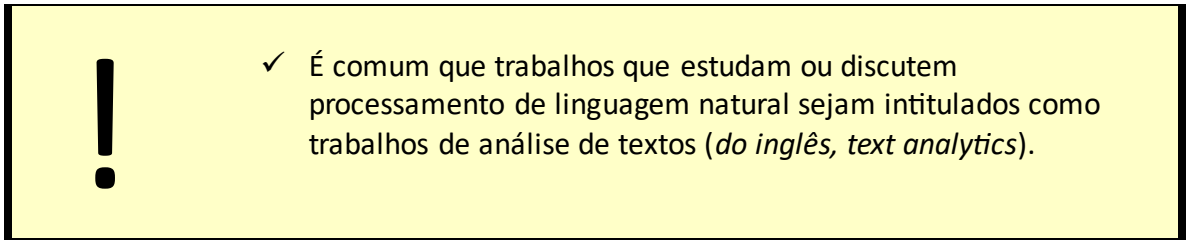
- ✓ **Volume de Armazenamento:** refere-se ao espaço físico ocupado pelos dados acessados, criados ou manipulados pelos sistemas computacionais. Bases de dados contendo dados de usuários em geral são fisicamente volumosas.
- ✓ **Volume decorrente de Variedade**: refere-se à grande quantidade de possibilidades para certas categorias de dados. Exemplos de dados com elevada variedade são aqueles presentes em textos de obras literárias.
- ✓ **Volume decorrente de Velocidade de Geração**: possui relação com dados que são gerados com elevadas frequências. Exemplos incluem dados gerados por sensores que executam tarefas periódicas.

Para o caso de dados no formato de linguagem natural (i.e., linguagem escrita, nesse caso), os dados requerem elevado armazenamento, são muito variados e são gerados em elevadas velocidades.

Dois dos desafios que surgem dizem respeito ao armazenamento desses dados e sua análise.

No que se refere à análise, a principal dificuldade diz respeito à falta de estruturação dos dados provenientes da linguagem natural.

A área de Processamento de Linguagem Natural lida justamente com a análise de tais dados.



Outro aspecto marcante nos dias atuais tem relação com a necessidade de automação de tarefas sem no entanto comprometer a interação humano-computador. Nesse caso, isso refere-se à **automatização de tarefas e melhoria na interação humano-computador (IHC)**.

Alguns exemplos típicos incluem o pré-atendimento efetuado em sistemas de triagem, a resolução de problemas e dúvidas recorrentes em sistemas de ajuda ao usuário, entre outros.

Para esse caso, o desafio é manter uma interface de interação com o ser humano que esteja em um nível de comunicação o mais próximo possível daquele com o qual esse está acostumado.

Desenvolver sistemas capazes de interagir em linguagem natural exigem menos esforços por parte do ser humano, durante a interação, tornando a comunicação menos susceptível a ruídos e acarretando maior êxito no processo de interação.

Cabe ressaltar, no em tanto, que elevar o nível da interação da máquina obviamente gera ao desenvolvedor do sistema computacional maior complexidade e transfere os riscos do processo comunicativo à máquina.

1.2. DEFINIÇÕES FUNDAMENTAIS

No estudo de PLN, os seguintes conceitos mostram-se relevantes para um apropriado entendimento da área:

PROCESSO COMUNICATIVO

É o processo que envolve a transmissão de uma **mensagem** partindo do **transmissor** e chegando ao **receptor**.

O transmissor gera uma referência, utilizando um conjunto de **símbolos (códigos)**, que representam um **referente** (e.g. realidade, fato, etc.). O receptor recebe os símbolos e, através do processo cognitivo, consegue **decodificar** inferir o referente.

LINGUAGEM

É o meio de se expressar os sinais utilizados no processo comunicativo.

A linguagem pode ser falada, gestual, escrita, gráfica, etc.

Dependendo da linguagem utilizada, o processo comunicativo pode ser mais ou menos facilitado.

A ciência que estuda a linguagem é a **linguística**.

✓ Outro termo comumente utilizado para referenciar trabalhos da área de PLN é linguística computacional.

LINGUAGEM NATURAL

É aquela que foi criada e que **evoluiu gradativamente** através do uso diário pelos seres humanos.

A linguagem natural é **dinâmica e temporal**.

Difere da linguagem artificial criada com um propósito específico (e.g.: SQL, linguagem de programação, etc.).

PROCESSAMENTO DE LINGUAGEM NATURAL

É um ramo especializado da **computação/engenharia**, originário da linguística computacional. Tem relação direta com a área de interação humano-computador (IHC) e foca no estudo e construção de sistemas que permitam a **interação de máquinas com as linguagens naturais** criadas pelos humanos.

SINTAXE

No processo comunicativo, sintaxe se refere às regras que regem a construção dos elementos básicos de uma determinada linguagem (palavras, frases, etc.).

Apesar de simples para um ser humano, efetuar a análise sintática de uma linguagem não é tarefa fácil sob o ponto de vista computacional.

O conjunto de palavras e expressões de uma língua é denominada de **léxico.**

A **morfologia** estuda as classes de palavras e, também, como estas são construídas a partir das suas menores unidades no contexto de uma língua. Essas unidades são denominadas de morfemas.

Exemplo: infeliz → morfema **radical** feliz + **prefixo** in morfema com sentido de "não".

A morfologia das palavras é aplicada ao PNL em técnicas tais como lematização (***lemmatization***) e stemização (***stemming***).

SEMÂNTICA

Refere-se aos significados atribuídos aos símbolos ou sinais de uma determinada linguagem.

A análise semântica é uma das tarefas mais difíceis para sistemas computacionais de PLN, pois envolve modelar e executar tarefas que envolvem aspectos cognitivos.

No processo de rotulação de elementos constituintes de uma sentença, escrita em uma língua, a análise semântica e sintática geralmente acarretam a obtenção de **rótulos** (***tags***) para as palavras (e.g. Verbo, substantivo/nome, adjetivo, advérbio, artigo, etc.)

CORPUS DE TEXTO (*TEXT CORPUS*)

É um conjunto de dados linguísticos reais, pertencentes a uma língua, que pode ser utilizado pelo computador.

O plural é **corpora de texto (*text corpora*)** e corresponde a um conjunto grande de *corpus*. São utilizados para análise estatística e construção de sistemas de processamento de linguagem natural.

Tipicamente os *corpora* são providos de metadados que facilitam o processamento computacional. Exemplos de metadados incluem:

- ✓ **POS tagging:** relaciona uma palavra a uma parte do discurso (*Part Of Speech*) e a rotula segundo sua função (e.g.: verbo).
- ✓ **Stemming tagging**: remoção de afixos de uma palavra sem necessariamente se chegar a um significado. Tem como foco a palavra base.
- ✓ **Lemma tagging:** Similar ao stemming, mas considera a redução à uma forma base preservando o significado.
- ✓ **Tipos semânticos e papeis:** anotações com vários elementos constituintes de uma sentença, incluindo palavras e frases.

TREE BANKS

São corpora especializados que contém formas avançadas de anotações sintáticas e semânticas. Esses corpora são denominados de treebanks.

Exemplo: WordNet, criado pela Universidade de Princeton em 1985, forcado na língua inglesa, que contém sinônimos (synsets), definição de palavras, relacionamentos, etc.

LEXEMA E LEMA

Um **lexema** é um conjunto de palavras de uma mesma **classe morfológica**. A forma base dessas palavras é denominada de **lema** (***lemma***)

Lema	Lexema
flor	florescer
	florido
	floresce

OBSERVAÇÃO: A técnica de ***stemização***, diferentemente da ***lematização***, ao eliminar afixos de uma palavra, pode chegar a uma estrutura raiz que não necessariamente é provida de significado ou correção gramatical.

Stem	Palavras originais
bol	bolacha
	boliche
	bolinha

1.3. ALGUMAS ÁREAS DE APLICAÇÃO DE PLN

A seguir, são ilustradas algumas áreas que constituem aplicação de processamento de linguagem natural:

TRADUÇÃO AUTOMATIZADA DE TEXTO

A tradução automatizada de textos é uma das aplicações mais difundidas na Internet.

Apresenta vários desafios, entre eles a existência de diferentes estruturas presentes em idiomas distintos, incluindo regras de sintaxe e semântica. A seguir, ilustra-se um exemplo de dificuldade experimentada por um sistema automatizado de tradução:

Frase em português: “Que cachorro bonito!”

Frase em inglês: “What a beautiful dog!”

Claramente a inversão de ordem entre o substantivo (cachorro) e o adjetivo (bonito), quando se comparam os dois idiomas, ilustra uma clara dificuldade enfrentada pelo processo de tradução automatizada.

SISTEMAS DE RECONHECIMENTO DE FALA

Uma outra área que tem ganhado importância em especial com relação à acessibilidade.

Esses sistemas geralmente são dependentes de domínio (i.e. um sistema que funciona bem para um pré-atendimento em uma oficina mecânica não necessariamente funcionará bem para uma triagem médica) e também são tipicamente customizados de acordo com as necessidades de um determinado usuário.

Aspectos relacionados ao sotaque regional, entonação, entre outros, constituem evidentes dificuldades experimentadas por esses tipos de sistemas.

SISTEMAS DE PERGUNTAS E RESPOSTAS

São muito comuns em sistemas de triagem textual, tais como chatbots, etc.

Uma dificuldade no desenvolvimento desses sistemas está na manutenção de uma base de conhecimento que seja vasta o suficiente para um determinado fim e, também, passível de consulta apropriada.

Assim como no caso dos sistemas de reconhecimento de voz, possui especificidade com relação às bases de conhecimento (e.g.: manutenção mecânica, saúde, etc.).

Exemplos típicos de desafios para esses tipos de sistema são a necessidade de robustez à escrita incorreta, tempo de resposta rápido, etc.

RESOLUÇÃO E RECONHECIMENTO CONTEXTUAL

Envolve a determinação do sentido de uma palavra que pode ter mais de um significado ou referente.

Exemplo: "João chamou André. Ele estava ocupado."

Nesse caso, o pronome "Ele" refere-se a quem? Notar que isso depende do contexto maior onde as frases estão inseridas e, sob o ponto de vista computacional essa determinação nem sempre é simples.

SUMARIZAÇÃO DE TEXTO

Lida com a redução do conteúdo de modo apropriado para criar um sumário/resumo que contenha os principais pontos do corpus de texto.

A similaridade entre tópicos abordados ou sua elevada diversidade tornam essa tarefa bastante desafiadora.

Os sumarizadores de texto podem ser agrupados em duas grandes categorias:

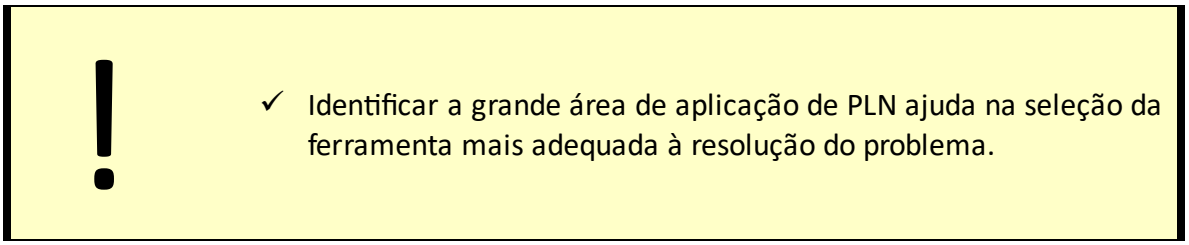
- **Sumarizadores extrativos:** são caracterizados por um subconjunto de palavras, frases, orações ou parágrafos que podem ser considerados representativos em relação ao texto original.
- **Sumarizadores abstrativos:** operam através da reescrita do texto original de modo a torná-lo menos volumoso sem perder a representatividade associada ao significado. Tipicamente parafraseiam um texto original são caracterizados por algoritmos avançados de aprendizagem de máquina.

CATEGORIZAÇÃO DE TEXTO

Uma aplicação também bastante difundida que lida com enquadramento de um corpus a uma determinada classe, baseado no seu conteúdo.

Exemplos de sistemas práticos incluem filtros de spam, análise de sentimentos, etc.

A categorização de texto aplicada à análise de sentimentos constitui uma poderosa ferramenta, utilizada pelas empresas, para alinhamento de seus serviços às expectativas de seus clientes.



1.4. PRINCIPAIS ETAPAS ENVOLVIDAS EM PLN

Para facilitar o estudo e a implementação de sistemas computacionais, o processamento de linguagem natural pode ser subdividido em diferentes etapas. Cada etapa pode ligada à etapa antecedente para receber dados fornece dados à etapa subsequente, tal como uma série de canos interconectados em um sistema de tubulação. Daí surgiu o termo “pipeline” de processamento.

A figura seguinte ilustra, em alto nível, as principais componentes dessa pipeline:

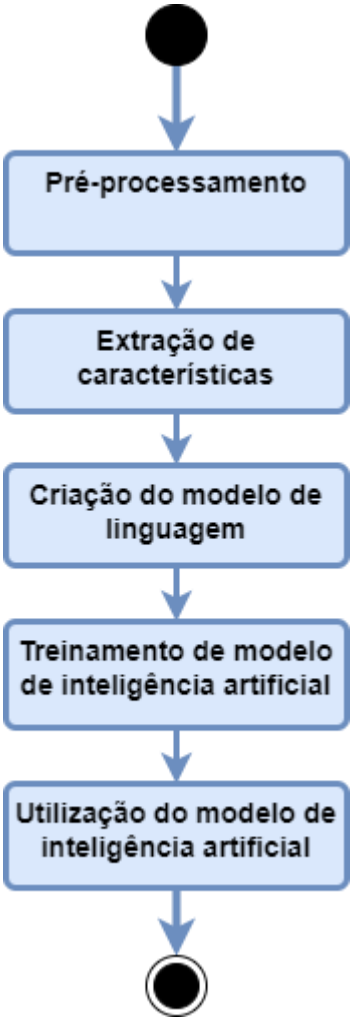


Figura 1. Elementos básicos de uma pipeline de PLN.

Na etapa de pré-processamento, os dados de linguagem natural são tratados de modo a se ter uma entrada livre de ruídos e relativamente padronizada.

A etapa de extração de características é a responsável por selecionar quais dados, da entrada padronizadas, serão utilizados na montagem do modelo.

O modelo de linguagem corresponde à representação numérico-matemática que será associada aos elementos da linguagem.

A etapa de treinamento de modelo de Inteligência artificial faz uso da representação do modelo de linguagem para treinar um modelo específico de IA voltado a uma certa tarefa.

Por fim, o modelo de IA, anteriormente treinado e devidamente salvo, é carregado e utilizado na execução da tarefa específica de PLN.

EXERCÍCIOS E PROBLEMAS:

TERMINOLOGIA E CONCEITOS

TC.1.1. Selecione uma obra literária de domínio público (ex. livros tais como Vinte mil léguas submarinas (de Júlio Verne), a Bíblia, etc.) e ilustre a variedade de dados presente. Considere, por exemplo a construção de frases, orações etc. e compare com expressões de uso corrente. Para respaldar sua resposta, elabore um programa que contabilize, por exemplo, o número de palavras diferentes.

Hint: Utilizar obras mais antigas, textos infantis, etc. Pode ajudar você a ter insights relacionados a tal variedade. Considere textos de tamanho equivalente ou calcule um índice que permita dar a noção de variedade (ex: taxa de palavras distintas utilizadas em relação ao total de palavras do texto, para textos de tamanhos similares).

TC.1.2. Exemplifique uma sentença, escrita na língua portuguesa, que pode surgir em um site de pré-atendimento em uma concessionária, que potencialmente seja difícil de ser interpretada por um chatbot. Explique sua resposta em termos de estruturação da sentença e suponha que ela esteja gramaticalmente correta. Descreva detalhadamente qual a dificuldade observada e como tal dificuldade poderia ser contornada reescrevendo-se a sentença.

TC.1.3. Sistemas de PNL são geralmente compostos por modelos que são treinados utilizando corpora de texto. Por que modelos que são válidos hoje podem não mais ser adequados daqui a dois anos? Dê um exemplo típico utilizando dados linguísticos apropriados.

TC.1.4. Por que a utilização de emojis ou outros símbolos não presentes na linguagem textual formal podem dificultar a operação de um sistema de PNL?

TC.1.5. Como o problema de utilização de emojis, citado no exercício **TC.1.4.** poderia ser resolvido na prática em um sistema de PLN?

TC.1.6. Dê um exemplo de sentença em um processo comunicativo onde a os referentes considerados pelo transmissor e pelo receptor podem ser distintos caso não haja adequada contextualização do processo comunicativo.

TC.1.7. Exemplifique uma saída para o processo de lematização e stemização. Considere as seguintes sentenças:

Sentença 1: "Assim que amanheceu, os estudantes, apressados, acordaram e saíram correndo para fazer a prova".

Sentença 2: "Os alunos da disciplina processamento de linguagem natural nunca passam pela disciplina sem ter aprendido o básico da área. Saem, portanto, minimamente capacitados a atuar e se aprofundar na área.

TC.1.8. Cite dois possíveis usos das tags do tipo POS. Forneça exemplos com sentenças simples, expressas na língua portuguesa ou inglesa.

TC. 1.9. Para os sistemas abaixo, diga quais envolvem ou não PLN, justifique sua resposta:

a) Um sistema de triagem automatizado, via Whatsapp, utilizando em um hospital com pronto atendimento.

b) Um sistema de diagnóstico de defeitos em um automóvel, baseado na descrição textual dos problemas relatados pelo motorista.

c) Um sistema de geração automática de código em linguagem de programação a partir de um diagrama de blocos funcional.

d) Um sistema de consulta a uma base de dados utilizando linguagem padrão SQL.

e) Um sistema de reconhecimento de fala utilizado pela Alexa.

f) Um sistema de reconhecimento de gestos utilizando a língua de sinais.

g) Um sistema para conversão de linguagem de programação em Python para JS (Transpilador).

TC.1.10. Pesquise na Internet algumas revisões de produtos e verifique a existência de sumários extrativos e abstrativos. Mostre o exemplo de sumário gerado. Adicionalmente, escolha um produto contendo um número considerável de revisões e escreva um sumário extrativo e um sumário abstrativo.

TC.1.11. Escreva quatro frases que podem apresentar problema de resolução e reconhecimento contextual. Especifique quais são esses problemas.

TC.1.12. Cite e descreva, em alto nível, dois modelos de linguagem e dois modelos de inteligência artificial.

PRÁTICA DE PROGRAMAÇÃO

PP.1.1. Baseando-se no código-fonte fornecido pelo professor, exemplifique o carregamento da biblioteca NLTK, em Python e efetue a tokenização de um texto em português pertencente a alguma obra literária de domínio público. Efetue uma tokenização por sentenças e uma tokenização por palavras. Qual a diferença de saída entre cada um dos processos?

Utilize um texto de pelo menos 2000 caracteres. Mostre o funcionamento do seu programa e descreva ao menos 5 POS tags.

PP.1.2. Exemplifique a stemização e a lematização de um texto, em língua portuguesa. Ilustre um caso onde textos diferentes conduzem a uma mesma saída através do stemming ou lemmatization. Considere como saída um vetor ordenado contendo lemas e stems.

PP.1.3. Repita **PP.1.1.** considerando a língua inglesa.

PP.1.4. Repita **PP.1.2.** considerando a língua inglesa.

PP.1.5. Repita **PP.1.2.** considerando a língua espanhola.

PP.1.6. Considere o seguinte problema: reescrever um texto em português ou inglês efetuando um mecanismo de substituição por sinônimos sem alterar o sentido de uma frase. Utilizando a biblioteca spacy ou nltk, desenvolva um programa, em Python, que recebe como entrada uma frase e retorna como saída uma frase reescrita utilizando sinônimos. Não é necessária a substituição de todas as palavras. Para a execução dessa tarefa, faça uso de informações presentes no corpora ou Treebank da linguagem.

PP.1.7. Desenvolva um programa em Python que lê um arquivo de entrada, em formato txt, quebra o texto em palavras e contabiliza quantas vezes a palavra ocorre no texto. Como saída o programa gera um gráfico que ilustra o número de vezes que cada palavra ocorreu. Em processamento de linguagem natural, como são chamadas as palavras que ocorrem com muita frequência e que podem não agregar muito significado? Para o texto que você analisou (que deve conter no mínimo 2000 palavras) quais seriam essas palavras? Quais as classes dessas palavras?

PP. 1.8. Repita o **PP.1.7.** mas utilizando um texto em inglês.

PP.1.9. Selecione ao menos 2 Treebanks da língua portuguesa e demonstre a instalação de eventuais pacotes, seu carregamento e utilização.

PP.1.10. Repita o **PP.1.9.** mas considerando o idioma inglês.

2. PRÉ-PROCESSAMENTO

Nessa seção, serão descritas algumas etapas de pré-processamento da pipeline de PLN.

O objetivo principal é habilitar o leitor a desenvolver componentes de pré-processamento aptos a viabilizar as etapas subsequentes da pipeline, em especial a etapa de extração de características que tipicamente sucede o pré-processamento.

O código-fonte que exemplifica várias das técnicas discutidas nessa seção podem ser encontrados em

https://github.com/fabriciogmc/natural_language_processing/tree/main/python/preprocessing

2.1. DEFINIÇÃO

A etapa de pré-processamento, também conhecida como etapa de **normalização** ou ***text wrangling*** é a primeira etapa de uma pipeline completa de PLN.

Nessa etapa, informações irrelevantes contidas do texto são removidas, são efetuadas conversões de caracteres em maiúsculas (ou minúsculas), palavras são reduzidas às suas formas básicas, são efetuadas correções ortográficas, são efetuadas expansões de abreviaturas e siglas, etc.

De um modo mais sistemático, podem ser listadas as seguintes etapas de pré-processamento básicas em uma pipeline:

- Remoção de marcações (e.g., tags HTML);
- Tokenização de sentenças ou frases;
- Tokenização de palavras;
- Remoção de caracteres especiais;
- Remoção de caracteres acentuados;
- Remoção de palavras de parada (***stop words***);
- Processamento de contrações, siglas e abreviaturas;

- Correção de erros de ortografia;
- Stemização (***stemming***);
- Lematização (***lematization***);
- Rotulação (***tagging***);
- Capitalização e descapitalização.
- Parsing;

Nas próximas seções serão descritas algumas dessas técnicas.

2.2. REMOÇÃO DE MARCAÇÕES

Essa atividade é executada principalmente após a atividade de coleta / raspagem de dados em rede (***web scraping)***.

Marcações (***tags***) e dados não informativos, tais como conteúdos de script, css, propagandas embarcadas em *iframes,* etc., são removidos do texto.

Se essa etapa não for executada, dados ruidosos e não informativos serão alimentados à etapa seguinte do pipeline.

2.3. TOKENIZAÇÃO

Pode ser definida como a quebra do texto em unidades menores (tokens) e mais significativas.

Tokens são unidades independentes que têm atreladas a si uma sintaxe e uma semântica bem definidas.

Tipicamente executa-se a tokenização por sentenças (frases, ***sentence tokenization***) e por palavras (***word tokenization***)

2.3.1. TOKENIZAÇÃO DE SENTENÇAS OU FRASES

Constituem um primeiro nível de quebra do corpus de texto.

Um documento é constituído por vários parágrafos que, por sua vez, podem ser divididos em uma ou mais frases (sentenças).

Algoritmos comuns de tokenização por frases incluem: busca por delimitadores específicos (ex. Pontos finais, ponto e vírgula, etc.), modelos de tokenização pré-treinados e tokenização através de expressão regular (***regex***).

2.3.2. TOKENIZAÇÃO DE PALAVRAS

Consiste na quebra de uma frase em palavras que a constituem.

Tipicamente a saída desse processo é utilizada como entrada para a limpeza e normalização dos dados, diretamente relacionada com a stemização e lematização.

2.4. REMOÇÃO DE CARACTERES ESPECIAIS

Essa etapa é necessária quando a acentuação não agrega significado à expressão ou termo. Nesse caso, acentos que são utilizados somente para enfatizar uma sílaba, por exemplo no caso da língua falada, não precisam ser mantidos.

Cabe ressaltar, entretanto, que em idiomas tais como o português, o acento distingue de modo significativo certas palavras, alterando o sentido da frase ou expressão.

Exemplos:

Ele é um bom estudante. (é – verbo)

Ele vai à aula e ele também dorme (e – conjunção coordenativa aditiva)

2.5. REMOÇÃO DE CARACTERES ESPECIAIS

Geralmente são caracteres não alfanuméricos que acrescentam ruído ao texto não estruturado.

Exemplos típicos são sinais de pontuação tais como exclamações, interrogações, etc.

Em alguns contextos, números podem também não agregar muita informação.

Cuidado! Caracteres especiais podem auxiliar a etapa de tokenização. Portanto, só devem ser removidos em etapas subsequentes!

Emojis, em alguns casos, podem ser considerados como conjuntos de caracteres especiais sujeitos à remoção. Em outras situações, podem ser tratados da mesma forma que siglas, contrações e abreviaturas.

2.6. PROCESSAMENTO DE CONTRAÇÕES, SIGLAS E ABREVIATURAS

Contrações, abreviaturas, siglas, etc., podem representar dificuldade ao sistema de PLN caso não sejam devidamente tratadas

Exemplos:

Copo d’água = Copo de água. (contração d'água)

Ltda = Limitada (abreviatura)

PLN = Processamento de Linguagem Natural (sigla)

Tipicamente abreviaturas, contrações, siglas, etc., são substituídas utilizando-se expressões regulares que operam baseadas em um conjunto de padrões mapeados por dicionários.

Exemplo:

```
{"ltda": "limitada", "pln": "processamento de linguagem natural", "d'água": "de agua"}
```

2.7. CORREÇÃO DE ERROS DE ORTOGRAFIA

Tem como objetivo a substituição de palavras que foram incorretamente escritas considerando-se a ortografia considerada correta.

Algumas vezes palavras podem ser escritas de modo incorreto para expressar uma intensidade ou um sentimento. Outras vezes podem ser escritas de modo incorreto por mero engano/descuido.

Exemplo: Faz muito friiiiiiiiiiiooooooooooooo.

Para o caso de padrões com letras repetidas, uma alternativa é proceder com a sucessiva remoção dessas, seguida de uma busca em um corpus de texto que pode ser utilizado como critério de parada.

Para o caso de grafia incorreta onde pode ocorrer múltiplas correspondência, a alternativa e utilizar uma medida de distância entre a palavra incorreta e outras candidatas, obtidas por alterações nesta.

Exemplo: pota

Substituições candidatas:

{porta, pata, pote, portal}

A substituição correta tipicamente leva em consideração uma análise da proximidade dos padrões de substituições em relação ao texto original e, também a ocorrência no corpus de texto utilizado como referência.

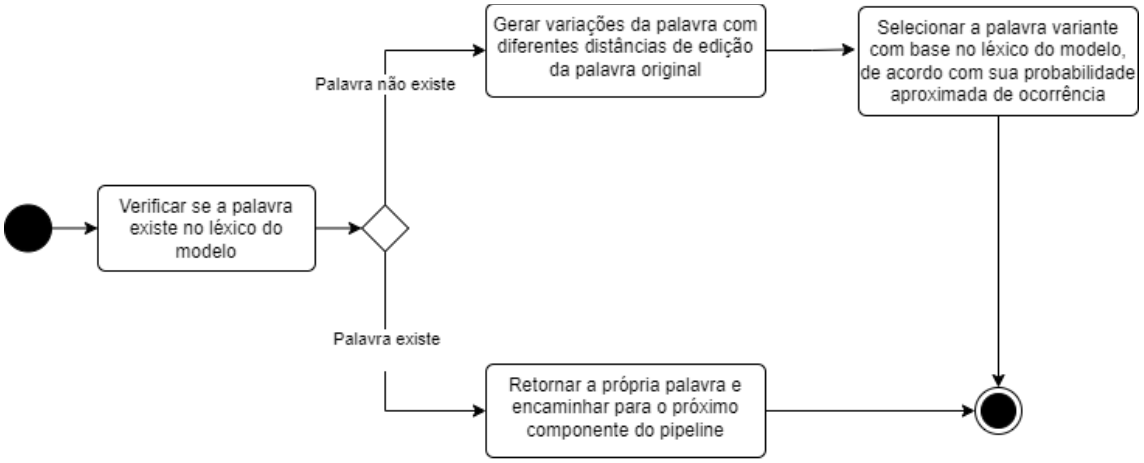
Exemplo: Análise de proximidade (***edit distance)***, considerando a palavra pota:

Porta – uma única operação de inserção entre o e t.

Portal – duas operações de inserção, entre o e t e ao final de da palavra.

O algoritmo de correção deve levar em consideração a probabilidade de ocorrência das substituições candidatas do corpus de texto tomado como referência para efetuar a correção.

Exemplo (cont): Esquematicamente, o mecanismo de correção pode ser implementado de acordo com o seguinte fluxo:



Exemplo: No caso do exemplo em questão, suponha-se que as palavras variantes geradas tenham as seguintes frequências de ocorrência no corpus de texto utilizado como referência:

porta	pote	portal	pata
2/100 = 2%	1/100 = 1%	5/100 = 5%	0/100 = 0%

Nesse caso, a palavra gerada escolhida para substituição seria portal

2.8. STEMIZAÇÃO

É a redução de uma palavra à sua forma raiz sem que haja, no entanto, a preservação do significado, ou seja, o stem não necessariamente faz parte do léxico da linguagem.

Exemplo: aborrecida → aborrec

Nesse caso, houve a eliminação do sufixo (característica do particípio) que foi agregado durante o processo de inflexão.

2.9. LEMATIZAÇÃO

É similar à stemização, mas, nesse caso, a palavra reduzida precisa fazer parte do léxico da linguagem.

Exemplos:

aborrecida → aborrecer

florada → flor

No processo de lematização, é comum que uma determinada classe gramatical seja reduzida a outra de mesma categoria durante o processo (e.g.: verbo flexionado → verbo no infinitivo, etc).

2.10. ROTULAÇÃO

A rotulação, conhecida como ***tagging,*** é a etapa de pré-processamento que identifica categorias de palavras presentes em uma frase de acordo com o contexto local.

A rotulação torna possível o estabelecimento de relações entre as diferentes palavras, substituição por palavras equivalentes, cálculos de estatísticas de frases específicas, entre outras.

Exemplo: O carro está sujo.

Saída do processo de *tagging*: `{o: artigo (DET), carro: substantivo (NN), está: verbo (VB), sujo: adjetivo (ADJ)}`

2.11. CAPITALIZAÇÃO E DESCAPITALIZAÇÃO

São utilizadas primordialmente para facilitar o processo de comparação e casamento de padrões de um ou mais padrões específicos.

Assim como no caso da remoção de caracteres especiais, textos providos de caracteres capitalizados podem auxiliar o processo de tokenização de sentenças e palavras.

2.12. REMOÇÃO DE PALAVRAS DE PARADA

Palavras de parada (***stop words***) são palavras que são muito frequentes em uma linguagem e que podem não agregar muito significado, prejudicando assim o processamento subsequente.

Exemplos:

Na língua portuguesa, tem-se artigos, preposições, etc., tais como {a, as, o, os, de, do, aquele, aquela}.

Observação: Algumas vezes, remover stop words pode alterar significativamente o sentido de uma frase e reduzi-la a outra totalmente diferente.

Exemplo de prejuízo ao pipeline de PLN após a remoção de uma hipotética *stop word,* considerando o seguinte conjunto de stop words sw = {eu, não, do}:

Frase original	Frase após remoção da stop word
Eu não gostei do computador	gostei computador
Eu gostei do computador	gostei computador

Nesse caso, frases com sentidos totalmente opostos acabaram por ser convertidas a expressões textuais equivalentes.

EXERCÍCIOS E PROBLEMAS:

TERMINOLOGIA E CONCEITOS

TC.2.1. Sob o ponto de vista linguístico, qual a diferença entre corpus de texto, parágrafo, frase, oração e palavra? Ilustre com um exemplo e indique como diferentes tipos de significado podem estar atrelados a cada um desses elementos.

TC.2.2. Qual um possível efeito da não remoção de um *iframe* ou de *scripts* e CSS's de um documento capturado através de uma raspagem de dados?

TC.2.3. Considerando que a tokenização corresponde à quebra do corpus de texto em elementos menores, exemplifique como diferentes quebras podem dar origem a diferentes estatísticas associadas a cada unidade menor constituinte (dê um exemplo numérico simples).

TC.2.4. Cite exemplos de **tokens de frases** e **tokens de palavras** que podem significar:

a) Uma opinião negativa referente a um produto de uma loja de vestuário;
b) Uma opinião positiva referente a um produto de uma loja de vestuário;
c) Uma opinião neutra referente a um produto de uma loja de vestuário;

d) Uma opinião negativa relacionada a um carro vendido por uma concessionária automotiva;
e) Uma opinião positiva relacionada a um carro vendido por uma concessionária automotiva;
f) Uma opinião neutra relacionada a um carro vendido por uma concessionária automotiva;

TC.2.5. Explique como a alteração do corpora de texto pode alterar o comportamento de um corretor ortográfico. Ilustre com uma frase simples que apresenta um erro de ortografia e que pode originar uma correção incorreta em virtude do processamento estatístico do corretor.

TC.2.6. Explique o que é uma expressão regular e como ela pode ser utilizada para projeto de corretor ortográfico para o caso de palavras com três ou mais caracteres repetidos.

TC.2.7. Cite três algoritmos de stemização e explique, em linhas gerais com um exemplo textual simples, como eles operam.

TC.2.8. Explique sistematicamente como você pode selecionar stop words para um corpora de texto quando você não tiver disponível uma biblioteca contendo uma lista de stop words. Utilizar bibliotecas contendo listas de stop words sem efetuar a análise que você citou pode acarretar algum problema prático para a pipeline?

PRÁTICA DE PROGRAMAÇÃO

PP.2.1. Desenvolva um sistema que faça a raspagem de dados de um site contendo opiniões de produtos. O sistema deve possuir uma interface onde o usuário informa uma determinada URL e um botão. A saída do sistema deve ser uma listagem contendo o texto plano das avaliações de um determinado produto.

PP.2.2. Desenvolva um sistema web que receba como entrada uma ou mais frases digitadas pelo usuário. O sistema efetua a tokenização por frases e, a seguir, efetua o tagging das classes gramaticais das frases e exibe, ao lado de cada frase listada, as palavras e seus rótulos, tais como o seguinte exemplo:

Entrada: Eu quero dormir.

Saída: {eu: pronome, quero: verbo, dormir: verbo}

PP.2.3. Tomando como base o código-fonte fornecido pelo professor e efetuando uma raspagem de dados que opere sobre alguma página contendo revisão de produtos, efetue uma comparação de desempenho utilizando 3 modelos de **tokenizadores de sentença**. Qual sua conclusão?

Obs: Na sua análise e resposta, contabilize o número de tokens gerados e explique a diferença. Meça também o tempo necessário para executar cada um dos processos de tokenização.

PP. 2.4. Descreva e exemplifique o comportamento de um tokenizador de sentença utilizando expressões regulares. Utilize um dado de revisão obtido no problema **PP.2.3.** (ou seja efetue a tokenização a partir de um dado recuperado de uma review da internet). Na sua resposta, mostre a quantidade dos tokens de saída após executar o programa e efetue uma validação para avaliar se o desempenho esperado é o obtido.

PP.2.5. Fazer o estudo comparativo de saída e desempenho dos **tokens de palavras** do nltk default, Treebank e Toktok. Utilize dados de uma review de **P.2.3.** No seu estudo, descreva, baseando-se nas saídas dos tokenizadores, quais as principais diferenças observadas quantificando os resultados com exemplos de textos simples.

PP.2.6. Exemplifique o funcionamento de um token de palavras utilizando expressão regular. Explique, com um programa exemplo, como configurar o padrão para obter um comportamento diferente do tokenizador.

PP.2.7. Tomando-se como base os problemas **TC.2.4** e **PP.2.3**, exemplifique a classificação de uma opinião (e.g., um comentário sobre um produto na Internet) pela mera **busca de um ou mais tokens de palavras**. Aplique as etapas de processamento ilustradas na aula de PLN. Na sua resolução mostre o seguite:

a) A obtenção dos tokens a partir da(as) sentença (as) origina(ais).

b) A identificação dos tokens que caracterizam a opinião positiva, negativa ou neutra.

c) A classificação da opinião baseada na mera ocorrência do token (faça um esquema em que um ou mais tokens podem ser utilizados para classificar a opinião).

d) Ilustre um caso onde esse mecanismo de classificação não funciona, mesmo que os tokens sejam encontrados.

PP.2.8. Exemplifique o funcionamento de um corretor ortográfico, aplicável à língua portuguesa, que efetue correção de palavras baseado em um corpus de texto considerado como referência e que utilize métricas de distância e estatísticas de ocorrência de palavras no corpus considerado.

Alterar o corpus pode afetar o comportamento do corretor? Se sim, dê um exemplo prático utilizando dados diferentes para o corpus.

A resposta a esse problema deverá ser um programa que:

a) Leia uma frase digitada pelo usuário.

b) Verifique se há ou não palavras potencialmente incorretas.

c) Informe ao usuário a frase potencialmente corrigida ou então diga que a frase aparenta estar correta.

PP.2.9. Aplique técnicas de tokenização e correção de erros de ortografia a dados de revisão de produtos que tenham sido raspados de uma página de revisão da Internet. Ilustre o comportamento e o desempenho do seu trecho de pipeline de PLN identificando os principais gargalos e sugira uma melhoria possível. Esclareça o porquê da ordem dos elementos em sua pipeline.

Na sua resposta ilustre, através de uma aplicação web, desenvolvida em Python, com front end e back end, os seguintes pontos:

a) O tempo necessário para se efetuar a raspagem de dados.

b) O tempo necessário à remoção de ruído

c) O tempo necessário à quebra em tokens de frases e tokens de palavras.

d) Substituição de abreviaturas

d) O tempo necessário para se processar e corrigir palavras digitadas incorretamente.

A entrada para seu programa deve ser uma URL.

As saídas devem ser cada uma das etapas citadas da pipeline de pré-processamento e os tempos necessários para se executar tais etapas.

Exemplo

Entrada	Saída	Etapa	Tempo
http://teste.com.br	“<html>...textos </html>”	Raspagem	5ms

“<html>...textos </html>”	“Produto interessantee. Gostei bastante. Mt!”	Remoção de ruído.	1ms
“Produto interessante. Gostei bastante. Mt!”	[‘Produto interessantee.’, ‘Gostei bastante.’, ‘Mt!”]	Tokenização por frases	2ms
[‘Produto interessantee.’, ‘Gostei bastante.’, ‘Mt!”]	[[‘Produto’, ‘interessantee’], [‘Gostei’, ‘bastante’],[‘Mt’]]	Tokenização por palavras	2.5ms
[[‘Produto’, ‘interessantee’], [‘Gostei’, ‘bastante’],[‘Mt’]]	[[‘Produto’, ‘interessantee’], [‘Gostei’, ‘bastante’],[‘Muito’]]	Expansão de siglas e abreviaturas	1ms
[[‘Produto’, ‘interessantee’], [‘Gostei’, ‘bastante’],[‘Muito’]]	[[‘produto’, ‘interessantee’], [‘gostei’, ‘bastante’],[‘muito’]]	Conversão para minúsculas	3ms
[[‘produto’, ‘interessantee’], [‘gostei’, ‘bastante’],[‘muito’]]	[[‘produto’, ‘interessante’], [‘gostei’, ‘bastante’],[‘muito’]]	Correção de caracteres incorretos	2.3ms

REFERÊNCIAS

[1] CARVALHO, F. G. M. Repositório com exemplos de programas para processamento de linguagem natural. Disponível em : <https://github.com/fabriciogmc/natural_language_processing.git>, Acesso em 26 de agosto de 2024.

[2] SARKAR, Dipanjan. Text analytics with python, 2nd ed, New York, Apress, 2019.

[3] LANE, Hobson; HOWARD, Cole; HAPKE, Hannes Max. Natural language processing in action. New York, Manning, 2019.