UERJ - UNIVERSIDADE DO ESTADO DO RIO DE JANEIRO
IME - INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
CComp - Programa de Pós-Graduação em Ciências Computacionais

IME999098 - Estágio de Docência I

Professora:
Rosa Maria Esteves Moreira da Costa

Apresentação sobre Processamento de Linguagem Natural (PLN)

Aluno: Roberto Carlos dos Santos
e-Mail: roberto.santos@pos.ime.uerj.br

14 de setembro de 2022

Processamento de Linguagem Natural (PLN)
Natural Language Processing (NLP)

Apresentação do tema

Processamento de Linguagem Natural (PLN)

Apresentação do tema
A comunicação entre máquinas e seres humanos

--> Seres humanos utilizam-se das linguagens naturais para se comunicar.

Autômato:
01101011 =>
Olá, humano!

Resposta do humano:
Olá, computador!
=> 00100100

Computadores e autômatos (robôs) necessitam de uma espécie de "tradução" (PLN) para a comunicação com humanos. -->

A seguir, exemplos de aplicações linguísticas de Classificação em PLN

Processamento de Linguagem Natural (PLN)
Natural Language Processing (NLP)

Exemplos de aplicações de PLN

- Sistemas de Diálogo
- Recuperação de informação e Extração de Informações
- Resposta a perguntas ou geração de perguntas
- Classificação de texto, discurso e imagem
- Reconhecimento de padrões
- Resumos automatizados de texto e de discursos
- Máquina de tradução
- Processamento de imagens

Processamento de Linguagem Natural (PLN)

Exemplos de aplicações de PLN - Sistemas de Diálogo

- Os **assistentes virtuais**: *Alexa* (Amazon), *Siri* (Apple), *Google Assistent* (Google) e *Cortana* (Microsoft);
- Os ***chat-bots*** existentes em portais de atendimento, de lojas virtuais, bancos, órgãos públicos na Internet;
- **Geração** e **análise** de diálogos;
- Diálogos **orientados** a metas ou tarefas de sistemas;
- Diálogos **visuais** – imagens e diálogos associados;
- Sistemas de **segurança** preventivos atuando em diálogos (previsão contra terrorismo, por exemplo).

Processamento de Linguagem Natural (PLN)

Exemplos de aplicações de PLN
Recuperação de informação e Extração de Informações

- ***Extração de informações***: consiste em **construir automaticamente** uma **base de conhecimento** (knowledge base-kb) estruturada lendo texto em linguagem natural.

A tarefa de construir automaticamente (ou "preencher") uma infobox a partir de texto é um exemplo de extração de informação. Grande parte da extração de informações pode ser descrita em termos de **entidades, relações e eventos**. [1. Definições de Eisentein]

- ***Recuperação da informação***: consiste em **encontrar material** (geralmente documentos) de uma **natureza não estruturada** (geralmente texto) que satisfaz uma necessidade de informação de grandes coleções (geralmente armazenadas em computadores)
  [2. definição dada por Christopher D. Manning et. alli]

Processamento de Linguagem Natural (PLN)

Exemplos de aplicações de PLN
Resposta a perguntas ou geração de perguntas

- Sobre assuntos de **domínio genérico**;
- Sobre assuntos de **domínio restrito**, como as relativas a uma área de conhecimento específico, como Geografia, Meteorologia, reservas de viagens etc;
- Relativas a **comunidades**, como *StackOverflow*, redes sociais etc;
- Resolução de questões de **múltipla escolha**, a partir da "leitura" de textos sobre o assunto;
- De **raciocínio lógico**;
- Relativas a determinadas **imagens**;
- Referentes à **similaridade** ou **reescrita** de questões;
- **Classificação e categorização** de questões;

Fonte principal:https://paperswithcode.com/area/natural-language-processing

Processamento de Linguagem Natural (PLN)

Exemplos de aplicações de PLN
Classificação de texto, discurso e imagem

- Classificação de **textos**, **documentos**, sentenças;
- Classificação de **sentimentos**, emoções, intenções ou predileções (útil, por exemplo, em serviços de streaming como *Youtube* e *Netflix*);
- **Modelagem** abstrata de **tópicos**, para a descoberta de estruturas semânticas ocultas em textos;
- Classificação de **relacionamentos**;
- Mineração de **argumentos**;
- Classificação de **reclamações**;
- Classificação de **vagas de trabalho**;
- Classificação de **imagens** (utilização de conceitos herdados, como os de *transformers, self-attention, embeddings of words* etc);

Fonte principal:https://paperswithcode.com/area/natural-language-processing

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN
Análise de sentimentos e de opiniões

Uma aplicação frequente de classificação de texto é determinar automaticamente o **sentimento** ou polaridade de **opinião** de documentos como análises de produtos e postagens em mídias sociais. Por exemplo, os profissionais de marketing estão interessados em saber como as pessoas respondem a anúncios, serviços; cientistas sociais estão interessados em saber como as opiniões e emoções se espalham pelas redes sociais, ou como são afetadas por fenômenos ambientais. A **análise de sentimentos** pode ser enquadrada como uma aplicação direta da **classificação de documentos**, assumindo que rótulos confiáveis podem ser obtidos. No caso mais simples, a análise de sentimento é uma problema de duas ou três classes, com sentimentos de POSITIVO, NEGATIVO e possivelmente NEUTRO. Essas anotações podem ser feitas à mão ou obtidas automaticamente.

Eisenstein, Jacob. Introduction to Natural Language Processing, 2018

Processamento de Linguagem Natural (PLN)

Exemplos de aplicações de PLN
Reconhecimento de padrões

- Mineração de padrões em **sequências**. Uso em Genética, p.e.;
- Detecção de discursos **inapropriados** em termos **éticos**, **morais**, **legais**;
- Detecção de **fraudes**, práticas **ilícitas** ou irregularidades. Por exemplos: plágios em textos, títulos enganosos;
- Detecção de **Spams e propagandas**;
- Detecção de ***fake news*** ou de perfis falsos;
- Detecção de **rumores, especulações** ou **boatos**;
- Detecção de **viéses** (tendenciosidades);
- Detecção de **erros** gramaticais;
- Detecção de **ironia**;
- Detecção de incongruências ou **incoerências** (em julgamentos, p.e.).

Fonte principal:https://paperswithcode.com/area/natural-language-processing

Processamento de Linguagem Natural (PLN)

Exemplos de aplicações de PLN
Resumos automatizados de texto e de discursos

- Sumarização de **documentos**, inclusive de coleção de documentos;
- Resumos curtos e concisos de textos;
- Geração de comentários automatizados em códigos-fonte de software;

Fonte principal:https://paperswithcode.com/area/natural-language-processing

Processamento de Linguagem Natural (PLN)

Exemplos de aplicações de PLN
Máquina de tradução

- Tradução de discursos em vídeos;
- Aplicativos de tradução de texto, como o Google Tradutor;
- Tradução automática não supervisionada;
- Tradução automatizada de linguagens técnicas (em Medicina e Direito, por exemplo);

Fonte principal:https://paperswithcode.com/area/natural-language-processing

Processamento de Linguagem Natural (PLN)

Exemplos de aplicações de PLN
Processamento de imagens

- **Detecção e reconhecimento** de imagens;
- Aumento artificial da base de dados, para treinamento (*image* **augmentation**);
- Descrição do conteúdo de uma imagem em palavras (*image* ***captioning***);
- **Geração de imagens** a partir de um texto.

Fonte principal:https://paperswithcode.com/area/natural-language-processing

Processamento de Linguagem Natural (PLN)
Natural Language Processing (NLP)

Exemplos de aplicações de PLN

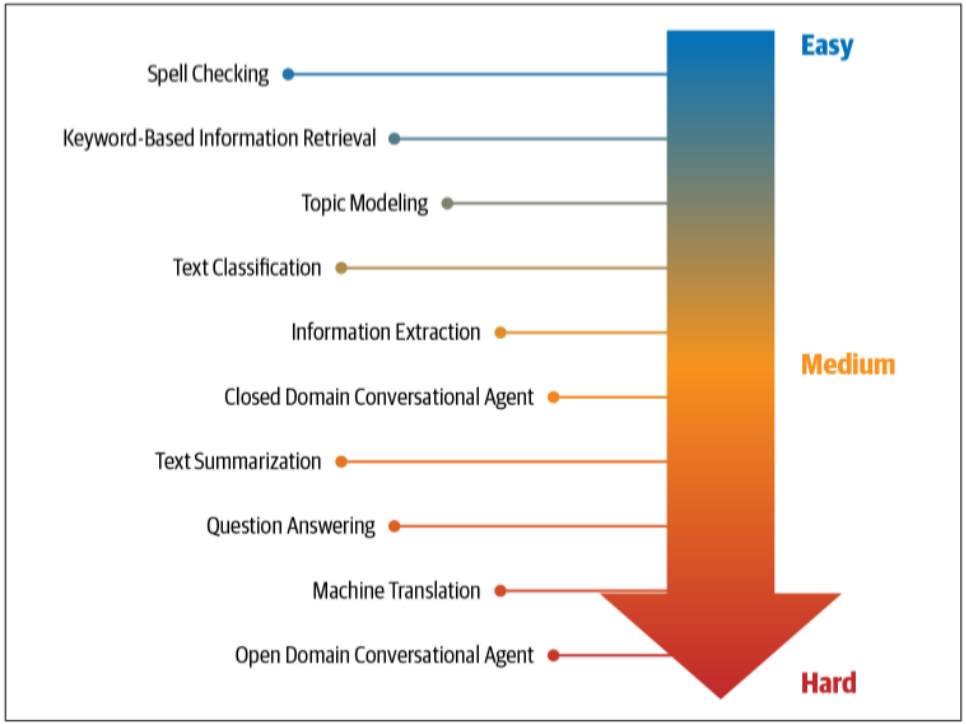


Figure 1-2. NLP tasks organized according to their relative difficulty

Practical Natural Language Processing, by Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana. 2020. Publisher(s): O'Reilly Media, Inc - ISBN: 9781492054054

Processamento de Linguagem Natural (PLN)
Natural Language Processing (NLP)

Exemplos de aplicações de PLN

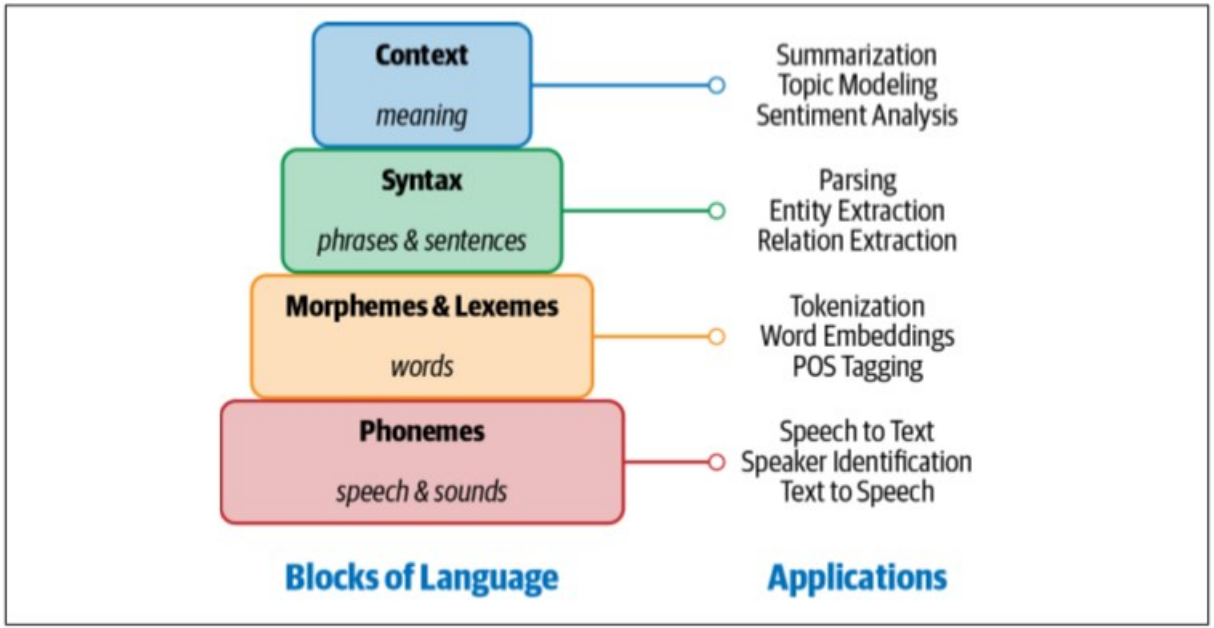


Figure 1-3. Building blocks of language and their applications

Practical Natural Language Processing, by Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana. 2020. Publisher(s): O'Reilly Media, Inc - ISBN: 9781492054054

Processamento de Linguagem Natural (PLN)

Ideias e pensadores que contribuíram para o avanço da PLN (exemplos)

Sem esgotar os exemplos, apresentaremos alguns pensadores e respectivas ideias ou obras que proporcionaram importantes resultados na evolução dos estudos da PLN.

Processamento de Linguagem Natural (PLN)

Ideias que contribuíram para o avanço da PLN

Leibniz e sua contribuição para as linguagens simbólicas

"[...] Leibniz, [...]percebeu [...] a conexão íntima entre a forma e o conteúdo da linguagem e as operações da mente humana. [...]
Estes pensamentos trouxeram a ideia de uma “linguagem universal”, uma linguagem artificial composta por símbolos que representasse conceitos por meio de regras lógicas e tornasse sua manipulação completamente válida. Leibniz acreditava que tal linguagem representaria perfeitamente os processos do raciocínio humano inteligível. [...] De acordo com Leibniz, a linguagem natural, apesar de ter poderosos recursos de comunicação, muitas vezes torna o raciocínio obscuro, já que não representa perfeitamente os pensamentos inteligíveis. [...]”.

Ferreira, Hugo Honda. Processamento de Linguagem Natural e Classificação de textos em Sistemas Modulares. Monografia apresentada como requisito parcial para conclusão do Bacharelado em Ciência da Computação. Orientador Prof. Dr. Flávio de Barros Vidal. Universidade de Brasília, 2019

Processamento de Linguagem Natural (PLN)

Ideias que contribuíram para o avanço da PLN

Alan Turing e seu clássico desafio (Turing Test)

Alan Turing escreveu, em 1959, um artigo denominado *Computing Machinery and Intelligence*. No tópico "*O jogo da imitação*", propôs a seguinte reflexão: *As máquinas podem pensar? Seriam capazes de imitar seres humanos de tal modo que um interlocutor não teria em média mais do que 70% de chances de fazer a correta identificação (sobre ser uma máquina e não um ser humano) depois de cinco minutos de diálogo (de questionamentos)?*

[Essas ideias serviriam mais tarde (atualmente) de inspiração para a criação de chat-bots e assistentes virtuais, por exemplo.]

M. Turing, Computing machinery and Intelligence, A. M. Turing (1950) Computing Machinery and Intelligence. Mind 49: 433-460

Processamento de Linguagem Natural (PLN)

Ideias que contribuíram para o avanço da PLN

Estruturas Sintáticas, de Noam Chomsky

"A PLN está profundamente entrelaçada com o estudo formal da linguagem, tanto conceitualmente quanto historicamente. Indiscutivelmente, essa conexão remonta ao pensamento de Chomsky publicado em sua obra ***Estruturas Sintáticas***, de 1957. Também vale hoje, com uma vertente de trabalhos recentes construindo análises formais de métodos modernos de redes neurais em termos de linguagens formais."

Formal Language Theory Meets Modern NLP
William Merrill - willm@allenai.org July 28, 2021

Seu sistema baseou-se em regras de como estruturar frases gramaticalmente corretas e isso inspirou muitas abordagens fundadas em regras sintáticas.

Revisão dos principais conceitos

Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos

Definição de *linguagem natural*

Uma ***linguagem natural*** pode ser definida como a linguagem utilizada por seres humanos para transmitir conhecimentos, emoções etc. A pessoa que deseja transmitir uma mensagem codifica-a usando sequências de sons ou de símbolos escritos, e a pessoa que a recebe decodifica a sequência para obter a mensagem original. A habilidade humana necessária para comunicar utilizando-se da linguagem natural é extremamente complexa e repousa sobre um processo que constrói estruturas complexas a partir de outras estruturas iniciais simples (símbolos escritos são utilizados para construir palavras, palavras são usadas para construir sentenças e sentenças são usadas para construir as mensagens a transmitir).

Araki, M., Delgado, R. L. C. (2007). Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment. Alemanha: Wiley.

Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos

Outra definição de linguagem natural

As linguagens naturais são linguagens faladas por humanos.
Linguagem natural é qualquer linguagem que os humanos aprendem em seu ambiente e usam para comunicarem-se uns com os outros.
Qualquer que seja a forma de comunicação, as linguagens naturais são usadas para expressar nossos conhecimentos e emoções e para transmitir nossas respostas a outras pessoas e ao nosso entorno.
As linguagens naturais são geralmente aprendidas na primeira infância a partir de aqueles ao nosso redor. Atualmente ainda não estamos no ponto em que essas linguagens em todas as suas formas não processadas podem ser compreendidas pelos computadores.

Reshamwala, Alpa & Mishra, Dhirendra & Pawar, Prajakta. (2013). REVIEW ON NATURAL LANGUAGE PROCESSING. IRACST – Engineering Science and Technology: An International Journal (ESTIJ). 3. 113-116.

Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos

Problemas da linguagem natural

"[...] sentenças em linguagem natural têm o potencial de serem sutis, complexas e repletas de ambiguidade, [...]."

Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In Proceedings of the 8th international conference on Intelligent user interfaces (IUI '03). Association for Computing Machinery, New York, NY, USA, 149–157. https://doi.org/10.1145/604045.604070

Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos

Problemas da linguagem natural

- Uma palavra pode ter **muitos significados** possíveis em diferentes contextos e resolver **ambiguidade** corretamente pode depender de muitos conhecimentos adicionais.
- **Riqueza**: qualquer significado pode ser expresso de muitas maneiras, e há imensuravelmente **muitos significados**.
- **Diversidade** linguística entre idiomas, dialetos, gêneros, estilos, domínios específicos ...
- A adequação de uma **representação** depende da aplicação.
- Qualquer representação é uma **construção teorizada**, não diretamente observável.
- Existem muitas fontes de **variação e ruídos** na entrada ling ca."

Natural Language Processing (CSEP 517): Introduction & Language Models - Noah Smith – 2017 - University of Washington - nasmith@cs.washington.edu
March 27, 2017 (com adaptações nesta tela)

Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos

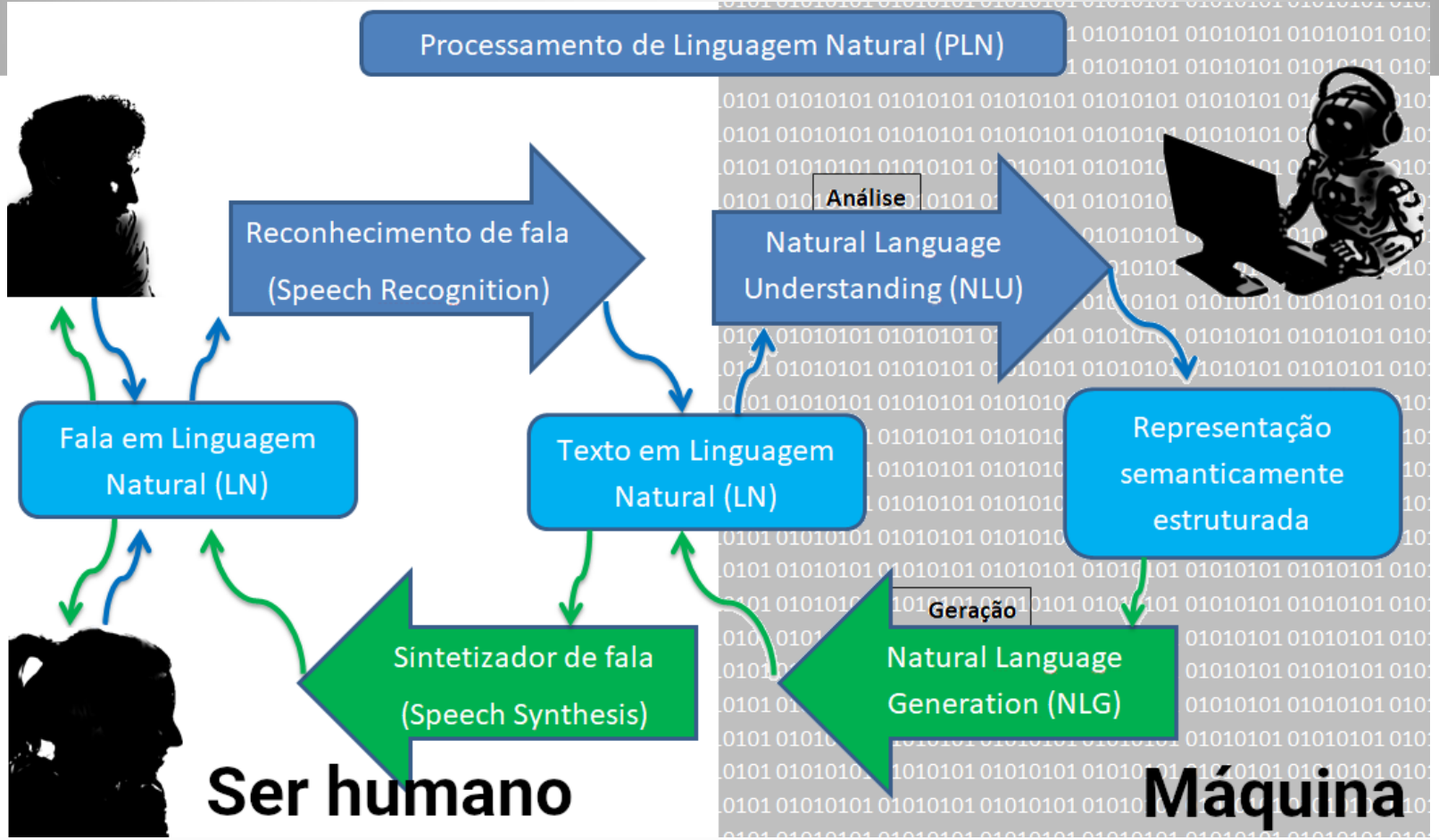
Problemas da linguagem natural

Exemplos de homônimos e homógrafos que podem gerar problemas no na compreensão do texto ou voz, até mesmo entre humanos:

- **Peça** a **peça**;
- **Passo** junto ao **Paço** e **sinto** o **laço** do **cinto lasso**;
- Tão logo a **Corte corte** as despesas;
- **Acordo cedo** para fazer o **acordo** e não **cedo**;
- Uso a **colher** para **colher** os frutos;
- **Coro** de vergonha em um **canto**, quando **canto** em um **coro**;
- Estava com muita **sede** quando cheguei na **sede** da empresa;
- **Gelo** quando carrego **gelo**;
- **Sobre** a **torre**, é possível que **sobre** algo que não se **torre**;
- **Alivio** no **começo** e **começo** a sentir **alívio**;
- **Bota** a **bota seca** e **seca** o **boto**;
- Sujou a **manga** com **manga**;
- O **Cabo** deu **cabo** (da situação) com um **cabo**;
- **Como** imaginam, **como** estou acima do peso, **como** bastante.

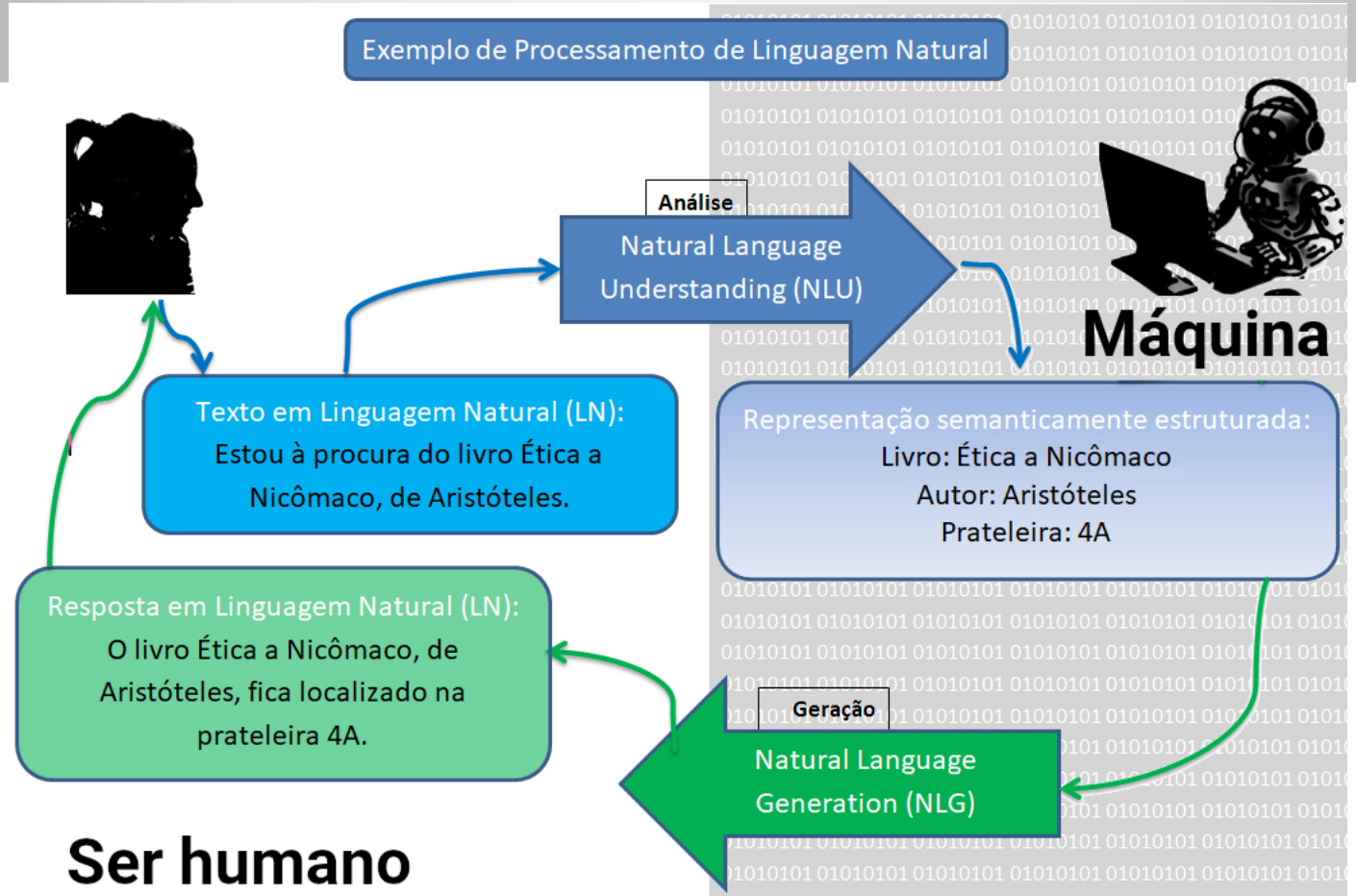
Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos



Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos



Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos

Algumas definições de PLN – Definição exemplificativa 1

PLN pode ser definida como um conjunto de técnicas para automatizar a habilidade humana de se comunicar utilizando-se de *linguagem natural*.

[Obs.: a definição de *linguagem natural* dessa mesma fonte encontra-se no Slide 4]

Araki, M., Delgado, R. L. C. (2007). Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment. Alemanha: Wiley.

Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos

Algumas definições de PLN – Definição exemplificativa 2

O ***Processamento de Linguagem Natural*** pode ser definido como a **síntese e o reconhecimento de voz** (Lange, 1993), e a **recuperação de informações** no cruzamento de linguagens (Oard & Diekema, 1998). A área de Processamento de Linguagem Natural (PLN) basicamente é utilizada para converter a linguagem humana em uma **representação reconhecível** ou em um formato que será fácil para os computadores utilizarem. Pode ser útil para desenvolver aplicações finais que atualmente incluem extração de informações, tradução automática, sumarização, pesquisa e interfaces de computador.

Tyagi, Garima and Singh, Rohit, Implementing CALL System using Natural Language Processing Tools (March 9, 2019). Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE) 2019, Available at SSRN: https://ssrn.com/abstract=3349556 or http://dx.doi.org/10.2139/ssrn.3349556

Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos

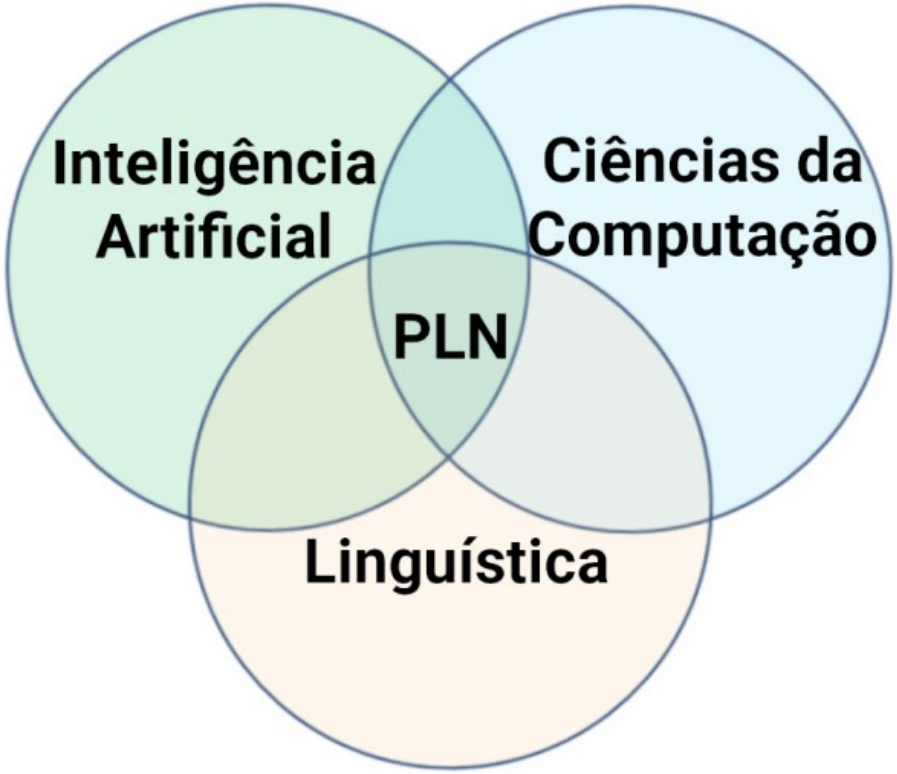
Algumas definições de PLN – Definição exemplificativa 3

O ***Processamento de Linguagem Natural*** pode ser definido como um campo de estudo que utiliza **Ciência da Computação**, **Inteligência Artificial e conceitos linguísticos** para analisar a ***linguagem natural***. Em outras palavras, A PNL é um conjunto de ferramentas usadas para derivar informações significativas a partir de dados textuais, e é geralmente utilizada para obter conhecimento e apoio à decisão através do processamento de dados textuais presentes nas páginas da Web, documentos, comentários de clientes.

Natural Language Processing for the Analysis Sentiment using a LSTM Model
BERRAJAA, Achraf.
International Journal of Advanced Computer Science and Applications;
West Yorkshire, Vol. 13, Ed. 5, (2022). DOI:10.14569/IJACSA.2022.0130589

Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos
Interseções com outras áreas de conhecimento



Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos

Algumas definições de PLN – Definição exemplificativa 4

O ***processamento de linguagem natural*** pode ser definido como a capacidade de um computador **processar linguagem** que humanos utilizam no **discurso comum** (como em inglês). Um objetivo primordial no processamento de linguagem natural é traduzir uma frase de entrada potencialmente ambígua em uma forma precisa que pode ser interpretada diretamente por um sistema de computador. Este processo de tradução, chamado *parsing*, é realizado de várias maneiras. [...] Os analisadores podem analisar sintaxe ou semântica ou ambos. A sintaxe se refere às regras que regem a ordem dos símbolos. A semântica, por outro lado, refere-se ao significado pretendido da expressão. Os computadores podem interpretar facilmente a sintaxe, mas são pobres em resolver semântica.

Optimization and Artificial Intelligence in Civil and Structural Engineering: Volume II: Artificial Intelligence in Civil and Structural Engineering Volume 221 de NATO Science Series E:
Editora Springer Science & Business Media, 2013
ISBN 940172492X, 9789401724920

Processamento de Linguagem Natural (PLN)

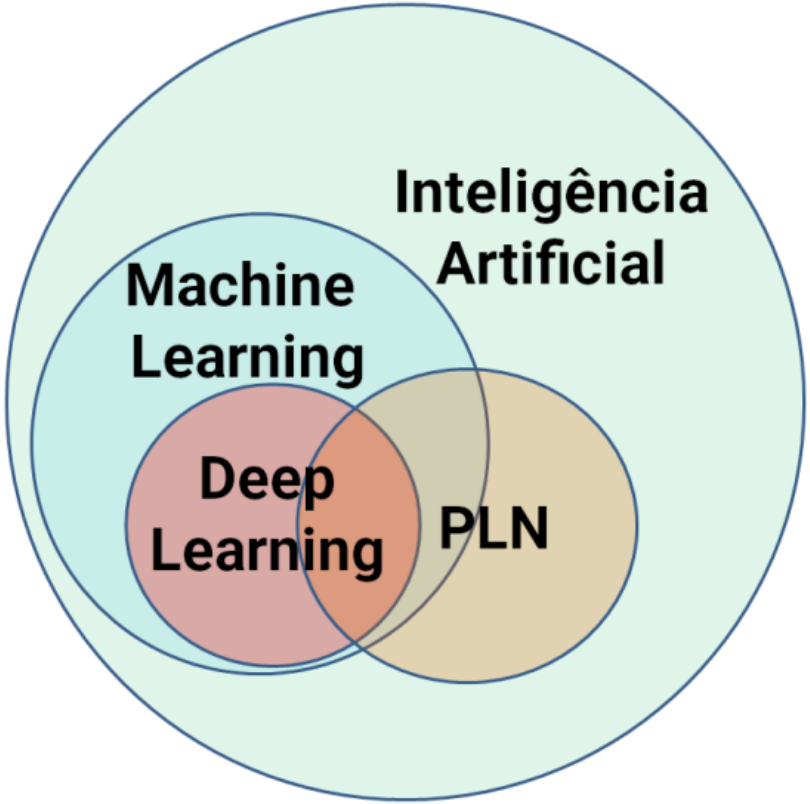
Revisão dos principais conceitos
Tentativa de sintetizar as definições de PLN vistas antes

PLN é a **área de estudo** interdisciplinar, que abrange **técnicas** de IA, Ciências da Computação e Linguística, que possibilita a mecanismos computadorizados **compreenderem e emularem** a **voz ou os textos** humanos, transformando-os em **representações** estruturadas para **processamento programado**, de modo a otimizar e simplificar a conversão das entradas (inputs) em dados ou processamentos úteis de saída (outputs) e a tornar a **interação entre usuário e máquina** mais assemelhada à existente entre seres humanos. A PLN é especialmente útil nos casos que envolvem o processamento de grandes quantidades de texto para a aquisição de informação.

Tentei, aqui, reunir as diversas definições em uma só, bastante abrangente.

Processamento de Linguagem Natural (PLN)

Revisão dos principais conceitos
Interseções no âmbito de Inteligência Artificial



Processamento de Linguagem Natural (PLN)

Técnicas e algoritmos frequentes em PLN

Veremos, a seguir, algumas das técnicas e algoritmos mais utilizados em PLN

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN
Corpus e corpora

Definições de ***corpus*** e de seu plural ***corpora*:**

- Coletânea de textos naturais, escolhidos para caracterizar um estado ou variedade de linguagem. (Sinclair, 1991).
- Corpo de linguagem natural (autêntica) que pode ser usado como base para pesquisa linguística. (Sinclair, 1991)
- Conjunto de dados linguísticos (pertencentes ao uso oral ou escrito da língua, ou a ambos), sistematizados segundo determinados critérios, suficientemente extensos em amplitude e profundidade, de maneira que sejam representativos da totalidade do uso linguístico ou de algum de seus âmbitos, dispostos de tal modo que possam ser processados por computador, com a finalidade de propiciar resultados vários e úteis para a descrição e análise (Sanchez, 1995)

Fonte: https://www.computersciencemaster.com.br/o-que-e-um-corpus-e-qual-sua-importancia-no-pln/

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN
Corpus e tokenização (*tokenization*)

Token – definição:
Cada "entidade" que faz parte do que quer que seja foi dividida com base em regras. Por exemplo, cada palavra é um *token* quando uma frase é "*tokenizada*" em palavras. Cada frase também pode ser um *token*, se você *tokenizar* as frases de um parágrafo.
Então, basicamente, *tokenizar* envolve separar frases e palavras do corpo do texto.
Na próxima tela, veremos exemplo de tokenização.

Fonte: https://acervolima.com/tokenizar-texto-usando-nltk-em-python/

Processamento de Linguagem Natural (PLN)

Corpus e tokenização (*tokenization*) - exemplo

```
import nltk
#nltk.download('stopwords')  #Basta fazer o download uma única vez. Depois, podem-se comentar estas linhas (esta e a próxima)
#nltk.download('punkt')
docs = [
    'A Corte decide o corte.',
    'Acordo cedo para o acordo e não cedo.',
    'Corte a colher para colher os frutos.',
    'Canto em um coro.',
    'Gelo quando carrego gelo.'
    ]
#O NLTK inclui stopword ("palavras limite") para o português:
stopwords = nltk.corpus.stopwords.words('portuguese')
#A esta altura, é possível utilizá-las para filtrar textos. Vamos encontrar as palavras mais
#comuns (à exceção das stopwords) e listá-las em ordem decrescente de freqüência:
# Conversão da coleção de sentenças em string
docs1=docs[0]+"\n"+docs[1]+"\n"+docs[2]+"\n"+docs[3]+"\n"+docs[4]
#Tokenização de sentenças
print(f"Tokens (%1d sentenças/tokens):" % len(tokens_sents))
tokens_sents = nltk.sent_tokenize(docs1)
print(tokens_sents)
print()
#Tokenização de palavras:
tokens = nltk.word_tokenize(docs1)
print(f"Tokens (%2d palavras/tokens):" % len(tokens))
print(tokens[:20])
print(tokens[20:33])
print("---")
# Filtrando as stop words e montando a distribuição de frequência
fd = nltk.FreqDist(w.lower() for w in tokens if w.lower() not in stopwords)
print("Frequência de tokens (em letras minúsculas, exceto stop-words):")
for word in fd.most_common():
  print (word)
print(f"Contagem de tokens distintos: %2d" % len(fd))
print()
```

https://colab.research.google.com/drive/1whPQvP-cUJnQPReYwP0qzx_1jiR3XEIM

Processamento de Linguagem Natural (PLN)

Corpus e tokenização (*tokenization*) - exemplo

```
Tokens (5 sentenças/tokens):
['A Corte decide o corte.', 'Acordo cedo para o acordo e não cedo.', 'Corte a colher para colher os frutos.', 'Canto em um coro.', 'Gelo quando carrego gelo.']

Tokens (33 palavras/tokens):
['A', 'Corte', 'decide', 'o', 'corte', '.', 'Acordo', 'cedo', 'para', 'o', 'acordo', 'e', 'não', 'cedo', '.', 'Corte', 'a', 'colher', 'para', 'colher']
['os', 'frutos', '.', 'Canto', 'em', 'um', 'coro', '.', 'Gelo', 'quando', 'carrego', 'gelo', '.']
---
Frequência de tokens (em letras minúsculas, exceto stop-words):
('.', 5)
('corte', 3)
('acordo', 2)
('cedo', 2)
('colher', 2)
('gelo', 2)
('decide', 1)
('frutos', 1)
('canto', 1)
('coro', 1)
('carrego', 1)
Contagem de tokens distintos: 11
```

https://colab.research.google.com/drive/1whPQvP-cUJnQPReYwP0qzx_1jiR3XElM

Processamento de Linguagem Natural (PLN)

Rotulagem de parte do texto
part of speech (pos) tagging

***Parts-of-speech (POS) Tagging* – definição:**
Rotulagem de partes da fala. Serve para analisar e identificar as diferentes classes gramaticais em um texto. A marcação de POS resulta em várias *tuplas*, onde cada uma delas contém a palavra e a sua *tag* (rótulo) que classifica gramaticalmente a palavra como verbo, adjetivo, substantivo, etc.

```
listaTaggedWords=nltk.corpus.mac_morpho.tagged_words()
#print(len(listaTaggedWords))
import random
for i in range(1,100,10):
    for j in range(1,10):
        nrAleat=random.randrange(1, 1170094)
        print(listaTaggedWords[nrAleat], end='')
    print("")
```

```
('por', 'PREP')('Ao', 'PREP')(',', ',')('todos', 'PROADJ')('provisória', 'ADJ')('por', 'PREP|+')('alma', 'N')('Luiz', 'NPROP')('e', 'KC')
('Foi', 'V')('de', 'PREP|+')('co-piloto', 'N')('por', 'PREP')('juros', 'N')('o', 'ART')('que', 'KS')('mergulhada', 'PCP')('Não', 'ADV')
('a', 'PREP|+')('mais', 'ADV')('"', '"')('entrever', 'V')('de', 'PREP')('veículos', 'N')('anúncio', 'N')('Teatro', 'NPROP')(',', ',')
('o', 'ART')('29,6', 'NUM')('em', 'PREP')('de', 'PREP|+')('a', 'ART')('tons', 'N')('por', 'PREP')('as', 'ART')('que', 'PRO-KS-REL')
('o', 'ART')('se', 'PROPESS')('prefixados', 'PCP')('Ilustrada', 'NPROP')('instrutiva', 'ADJ')(',', ',')('Uno', 'NPROP')('preferidos', 'N')('se', 'PROPESS')
('memorizou', 'V')('mas', 'KC')('Em', 'PREP|+')('?', '?')('86', 'N')('como', 'ADV-KS')('Zélia', 'NPROP')('pacientes', 'N')(',', ',')
('de', 'PREP')('o', 'ART')('&', 'NPROP')('redescoberta', 'PCP')('entediantes', 'ADJ')('que', 'PRO-KS-REL')('o', 'ART')(',', ',')('Samuel', 'NPROP')
('fixa', 'PCP')('depois', 'ADV')('geada', 'N')('de', 'PREP|+')('internacionais', 'ADJ')('efetuada', 'PCP')('referências', 'N')('conhecido', 'PCP')('com', 'PRE
('operacionais', 'ADJ')('e', 'KC')('Inteligência', 'NPROP')('o', 'ART')('internacional', 'ADJ')('Israelense', 'NPROP')('Suicídio', 'N')('que', 'KS')('consider
(',', ',')(',', ',')('que', 'KS')('é', 'VAUX')('trabalho', 'N')("'", "'")(',', ',')('ecológico', 'ADJ')('até', 'PREP')
```

https://www.alura.com.br/artigos/guia-nlp-conceitos-tecnicas
https://colab.research.google.com/drive/1oOdnZYnUVlX3H7PwYSmXdWfpzd8dWGIN?usp=sharing

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN
Remoção de "*stop words*"

As ***Stop Words*** são palavras que não necessitam ser indexadas, por possuírem pouco significado para a análise semântica de um texto, tais como preposições, artigos, conjunções e outros.
Reduzindo-se as palavras a analisar, reduz-se naturalmente a complexidade da análise.
Na tela a seguir, veremos a lista de palavras que a biblioteca NLTK considera como *stop-words*.

Exemplo de *Stop-Words*

Remoção de "*stop words*"

```python
import nltk
#nltk.download('stopwords')
stopwords = nltk.corpus.stopwords.words('portuguese')
#stopwords[15:30]
print(len(stopwords))
for i in range(1,207,9):
  print(stopwords[i-1:i+8])
```

```
207
['a', 'à', 'ao', 'aos', 'aquela', 'aquelas', 'aquele', 'aqueles', 'aquilo']
['as', 'às', 'até', 'com', 'como', 'da', 'das', 'de', 'dela']
['delas', 'dele', 'deles', 'depois', 'do', 'dos', 'e', 'é', 'ela']
['elas', 'ele', 'eles', 'em', 'entre', 'era', 'eram', 'éramos', 'essa']
['essas', 'esse', 'esses', 'esta', 'está', 'estamos', 'estão', 'estar', 'estas']
['estava', 'estavam', 'estávamos', 'este', 'esteja', 'estejam', 'estejamos', 'estes', 'esteve']
['estive', 'estivemos', 'estiver', 'estivera', 'estiveram', 'estivéramos', 'estiverem', 'estivermos', 'estivesse']
['estivessem', 'estivéssemos', 'estou', 'eu', 'foi', 'fomos', 'for', 'fora', 'foram']
['fôramos', 'forem', 'formos', 'fosse', 'fossem', 'fôssemos', 'fui', 'há', 'haja']
['hajam', 'hajamos', 'hão', 'havemos', 'haver', 'hei', 'houve', 'houvemos', 'houver']
['houvera', 'houverá', 'houveram', 'houvéramos', 'houverão', 'houverei', 'houverem', 'houveremos', 'houveria']
['houveriam', 'houveríamos', 'houvermos', 'houvesse', 'houvessem', 'houvéssemos', 'isso', 'isto', 'já']
['lhe', 'lhes', 'mais', 'mas', 'me', 'mesmo', 'meu', 'meus', 'minha']
['minhas', 'muito', 'na', 'não', 'nas', 'nem', 'no', 'nos', 'nós']
['nossa', 'nossas', 'nosso', 'nossos', 'num', 'numa', 'o', 'os', 'ou']
['para', 'pela', 'pelas', 'pelo', 'pelos', 'por', 'qual', 'quando', 'que']
['quem', 'são', 'se', 'seja', 'sejam', 'sejamos', 'sem', 'ser', 'será']
['serão', 'serei', 'seremos', 'seria', 'seriam', 'seríamos', 'seu', 'seus', 'só']
['somos', 'sou', 'sua', 'suas', 'também', 'te', 'tem', 'tém', 'temos']
['tenha', 'tenham', 'tenhamos', 'tenho', 'terá', 'terão', 'terei', 'teremos', 'teria']
['teriam', 'teríamos', 'teu', 'teus', 'teve', 'tinha', 'tinham', 'tínhamos', 'tive']
['tivemos', 'tiver', 'tivera', 'tiveram', 'tivéramos', 'tiverem', 'tivermos', 'tivesse', 'tivessem']
['tivéssemos', 'tu', 'tua', 'tuas', 'um', 'uma', 'você', 'vocês', 'vos']
```


https://colab.research.google.com/drive/1-3KSa2Xgh7_QB0v_IsSSaq2zw6eWef-I?usp=sharing

Classificação de texto linear

A classificação de texto tem muitas aplicações, desde a filtragem de *spam* até a análise de registros eletrônicos de saúde.

Classificação de texto linear
A Sacola de palavras (*Bag of Words*)

O modelo de sacola de palavras (*bag-of-words - BOW*) é uma **representação** que transforma texto arbitrário em **vetores** de comprimento fixo. Esses vetores podem conter, por exemplo, a contagem, a frequência, a indicação booleana da existência de cada palavra. Este processo é muitas vezes referido como vetorização. Veremos um exemplo a seguir.

Processamento de Linguagem Natural (PLN)

Classificação linear de texto
A Sacola de palavras (*Bag of Words*)

```python
from keras.preprocessing.text import Tokenizer

docs = [
  'No passado, a Corte decidiu pelo corte.',
  'Corte em peça inteira.',
  'Gosto bem passado',
  'Passo no canto.',
  'Estudo canto.',
  'Peça à direção para assistir à peça.',
  'Em direção ao canto, ouço o canto.'
]

## Passo 1: Determinação do vocabulário
tokenizer = Tokenizer()
tokenizer.fit_on_texts(docs)
print(f'Vocabulary: {list(tokenizer.word_index.keys())}')

## Passo 2: Contar palavras - mode  pode ser: "binary", "count", "tfidf", "freq".
print("Binary:")
vectors = tokenizer.texts_to_matrix(docs, mode='binary')
print(vectors)
print("----")
print("Count:")
vectors = tokenizer.texts_to_matrix(docs, mode='count')
print(vectors)
print("----")
print("Tfidf:")
vectors = tokenizer.texts_to_matrix(docs, mode='tfidf')
print(vectors)
print("----")
print("Freq:")
vectors = tokenizer.texts_to_matrix(docs, mode='freq')
print(vectors)
```

```
Vocabulary: ['canto', 'corte', 'peça', 'no', 'passado', 'em', 'à', 'direção', 'a', 'decidiu', 'pelo', 'inteira', 'gosto', 'bem', 'passo', 'estudo', 'para', 'assistir', 'ao', 'ouço', 'o']
Binary:
[[0. 0. 1. 0. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 1.]]
----
Count:
[[0. 0. 2. 0. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 2. 0. 0. 0. 2. 1. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0.]
 [0. 2. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 1.]]
```

https://colab.research.google.com/drive/1whPQvP-cUJnQPReYwP0qzx_1jiR3XElM?usp=sharing

Datasets de exemplo: https://archive.ics.uci.edu/ml/datasets/Bag+of+Words

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN
Desambiguação do sentido da palavra

Muitas frases podem ser ambíguas porque contêm palavras que têm múltiplos significados ou sentidos. A desambiguação do sentido da palavra é o problema de identificar o sentido pretendido de cada *token* de palavra em um documento. A desambiguação do sentido das palavras faz parte de um campo maior de pesquisa chamado ***semântica lexical***, que se preocupa com os significados das palavras.

Em um nível básico, o problema da desambiguação do sentido da palavra é identificar o sentido correto para cada *token* de palavra em um documento.

Eisenstein, Jacob. Introduction to Natural Language Processing, 2018

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN
Desambiguação do sentido da palavra (continuação)

A ambiguidade de parte do discurso (por exemplo, substantivo versus verbo) é geralmente considerada um problema diferente, a ser resolvido em um estágio anterior.
De uma perspectiva linguística, os sentidos não são propriedades de palavras, mas de ***lemas***, que são **formas canônicas** que representam um conjunto de palavras flexionadas. Portanto, a desambiguação do sentido da palavra requer primeiro identificar a parte da fala e o lema corretos para cada *token* e, em seguida, escolher o sentido correto do inventário associado ao lema correspondente.

Eisenstein, Jacob. Introduction to Natural Language Processing, 2018

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN
Derivação (*stemming*) e Lematização (*lematization*)

O objetivo tanto da derivação quanto da lematização é **reduzir** as formas flexionais e às vezes as formas derivadas de uma **palavra** a uma **forma básica comum**.

Na **derivação** (*stemming*), analisa-se cada palavra individualmente e a reduz-se à sua raiz (*stem*). Pode-se reduzir a palavra a uma outra gramaticalmente incorreta, porém ainda com valor para análise. Os algoritmos de *stemming* têm um conjunto de regras para decidir como fazer os cortes.

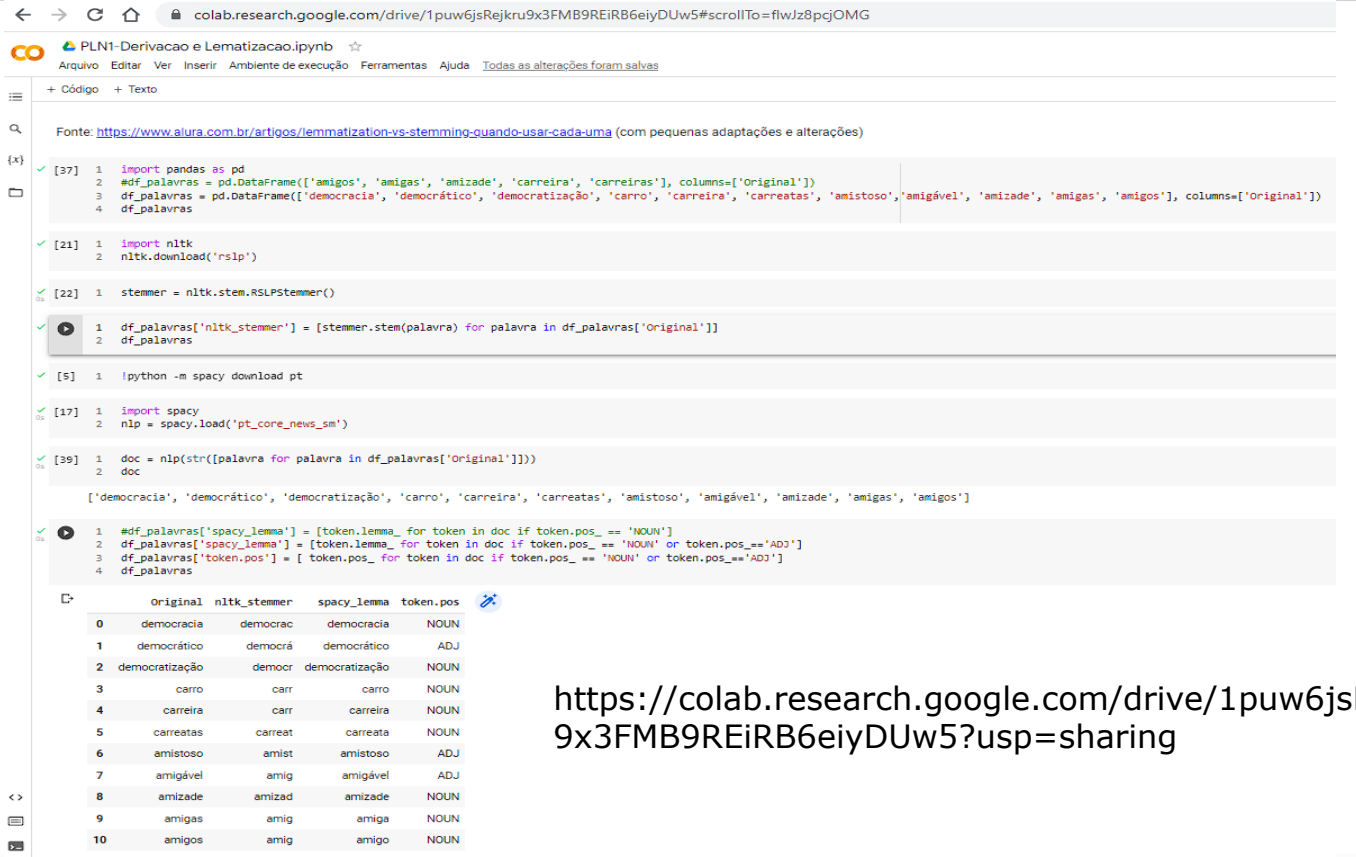
Na **lematização** (*lemmatization*) também se pode reduzir a palavra, retirando-se todas as inflexões e chegando-se ao seu lema. Porém, essa redução sempre resultará em uma palavra que realmente existe na gramática. Outro ponto importante é que, nessa técnica, a classe gramatical da palavra será levada em consideração para fazer a redução.

No exemplo a seguir, utilizaremos a biblioteca **NLTK** para a derivação e a **spaCy** para a lematização.

Fontes: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html e https://www.alura.com.br/artigos/lemmatization-vs-stemming-quando-usar-cada-uma

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN
Derivação e Lematização – exemplo prático



colab.research.google.com/drive/1puw6jsRejkru9x3FMB9REiRB6eiyDUw5#scrollTo=flwJz8pcjOMG

PLN1-Derivacao e Lematizacao.ipynb

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

+ Código + Texto

Fonte: https://www.alura.com.br/artigos/lemmatization-vs-stemming-quando-usar-cada-uma (com pequenas adaptações e alterações)

```
[37] 1 import pandas as pd
     2 #df_palavras = pd.DataFrame(['amigos', 'amigas', 'amizade', 'carreira', 'carreiras'], columns=['Original'])
     3 df_palavras = pd.DataFrame(['democracia', 'democrático', 'democratização', 'carro', 'carreira', 'carreatas', 'amistoso','amigável', 'amizade', 'amigas', 'amigos'], columns=['Original'])
     4 df_palavras
```

```
[21] 1 import nltk
     2 nltk.download('rslp')
```

```
[22] 1 stemmer = nltk.stem.RSLPStemmer()
```

```
     1 df_palavras['nltk_stemmer'] = [stemmer.stem(palavra) for palavra in df_palavras['Original']]
     2 df_palavras
```

```
[5]  1 !python -m spacy download pt
```

```
[17] 1 import spacy
     2 nlp = spacy.load('pt_core_news_sm')
```

```
[39] 1 doc = nlp(str([palavra for palavra in df_palavras['Original']]))
     2 doc
```

['democracia', 'democrático', 'democratização', 'carro', 'carreira', 'carreatas', 'amistoso', 'amigável', 'amizade', 'amigas', 'amigos']

```
     1 #df_palavras['spacy_lemma'] = [token.lemma_ for token in doc if token.pos_ == 'NOUN']
     2 df_palavras['spacy_lemma'] = [token.lemma_ for token in doc if token.pos_ == 'NOUN' or token.pos_=='ADJ']
     3 df_palavras['token.pos'] = [ token.pos_ for token in doc if token.pos_ == 'NOUN' or token.pos_=='ADJ']
     4 df_palavras
```

	Original	nltk_stemmer	spacy_lemma	token.pos
0	democracia	democrac	democracia	NOUN
1	democrático	democrá	democrático	ADJ
2	democratização	democr	democratização	NOUN
3	carro	carr	carro	NOUN
4	carreira	carr	carreira	NOUN
5	carreatas	carreat	carreata	NOUN
6	amistoso	amist	amistoso	ADJ
7	amigável	amig	amigável	ADJ
8	amizade	amizad	amizade	NOUN
9	amigas	amig	amiga	NOUN
10	amigos	amig	amigo	NOUN

https://colab.research.google.com/drive/1puw6jsRejkru9x3FMB9REiRB6eiyDUw5?usp=sharing

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN
Derivação e Lematização – exemplo prático

	Original	nltk_stemmer	spacy_lemma	token.pos
0	democracia	democrac	democracia	NOUN
1	democrático	democrá	democrático	ADJ
2	democratização	democr	democratização	NOUN
3	carro	carr	carro	NOUN
4	carreira	carr	carreira	NOUN
5	carreatas	carreat	carreata	NOUN
6	amistoso	amist	amistoso	ADJ
7	amigável	amig	amigável	ADJ
8	amizade	amizad	amizade	NOUN
9	amigas	amig	amiga	NOUN
10	amigos	amig	amigo	NOUN

https://colab.research.google.com/drive/1puw6jsRejkru9x3FMB9REiRB6eiyDUw5?usp=sharing

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN

N-Grams

***n-grams* (definição):**

n-gram é um tipo de modelo probabilístico usado para prever o próximo item de uma sequência na forma de um *modelo de Markov*. Em um contexto linguístico, o *n-grams* refere-se a uma sequência **n** de palavras.

Veremos um exemplo na próxima tela.

Fonte da definição: https://www.alura.com.br/artigos/guia-nlp-conceitos-tecnicas

Processamento de Linguagem Natural (PLN)

Aplicações linguísticas da Classificação em PLN

N-Grams

```
1  from nltk import ngrams
2  sentence = 'No estudo de programação de linguagem natural é importante conhecer as relações entre as palavras.'
3  for n in range(2,6):
4      n_grams = ngrams(sentence.split(), n)
5      print("")
6      print ("%2d-gram: " % (n))
7      for grams in n_grams:
8          print (grams)
```

```
2-gram:
('No', 'estudo')
('estudo', 'de')
('de', 'programação')
('programação', 'de')
('de', 'linguagem')
('linguagem', 'natural')
('natural', 'é')
('é', 'importante')
('importante', 'conhecer')
('conhecer', 'as')
('as', 'relações')
('relações', 'entre')
('entre', 'as')
('as', 'palavras.')

3-gram:
('No', 'estudo', 'de')
('estudo', 'de', 'programação')
('de', 'programação', 'de')
('programação', 'de', 'linguagem')
('de', 'linguagem', 'natural')
('linguagem', 'natural', 'é')
('natural', 'é', 'importante')
('é', 'importante', 'conhecer')
('importante', 'conhecer', 'as')
('conhecer', 'as', 'relações')
('as', 'relações', 'entre')
('relações', 'entre', 'as')
('entre', 'as', 'palavras.')

4-gram:
('No', 'estudo', 'de', 'programação')
('estudo', 'de', 'programação', 'de')
('de', 'programação', 'de', 'linguagem')
('programação', 'de', 'linguagem', 'natural')
('de', 'linguagem', 'natural', 'é')
('linguagem', 'natural', 'é', 'importante')
('natural', 'é', 'importante', 'conhecer')
('é', 'importante', 'conhecer', 'as')
('importante', 'conhecer', 'as', 'relações')
('conhecer', 'as', 'relações', 'entre')
('as', 'relações', 'entre', 'as')
('relações', 'entre', 'as', 'palavras.')

5-gram:
('No', 'estudo', 'de', 'programação', 'de')
('estudo', 'de', 'programação', 'de', 'linguagem')
('de', 'programação', 'de', 'linguagem', 'natural')
('programação', 'de', 'linguagem', 'natural', 'é')
('de', 'linguagem', 'natural', 'é', 'importante')
('linguagem', 'natural', 'é', 'importante', 'conhecer')
('natural', 'é', 'importante', 'conhecer', 'as')
('é', 'importante', 'conhecer', 'as', 'relações')
('importante', 'conhecer', 'as', 'relações', 'entre')
('conhecer', 'as', 'relações', 'entre', 'as')
('as', 'relações', 'entre', 'as', 'palavras.')
```


https://colab.research.google.com/drive/1ZQ2HhomSko3rJGKykbPTrNrN73td1zf9?usp=sharing

Processamento de Linguagem Natural (PLN)

Reconhecimento de nome de entidade

Name Entity Recognition (NER)

Um *corpus* pode ser preparado para a categorização e reconhecimento de nomes de entidades, como no exemplo abaixo, onde ORG= organização, LOC=local e MISC=miscelânia:

```
1  #NER - Name Entity Recognition
2  for ent in doc.ents:
3      print(ent.text, ent.label_)
```

```
Apple ORG
Reino Unido LOC
Carros MISC
São Francisco LOC
Londres LOC
Reino Unido LOC
```


https://colab.research.google.com/drive/1oOdnZYnUVlX3H7PwYSmXdWfpzd8dWGIN

Processamento de Linguagem Natural (PLN)

Aprendizagem supervisionada versus aprendizagem não supervisionada

O ***aprendizado supervisionado*** é um modo de aprendizado com dados de treinamento que requer **intervenção humana**. Os modelos de *aprendizado supervisionado* geralmente são capazes de alcançar excelentes níveis de desempenho, mas somente quando dados rotulados suficientes estão disponíveis.

Além disso, a construção, dimensionamento, implantação e manutenção de modelos precisos de *aprendizado supervisionado* leva tempo e conhecimento técnico de uma equipe de cientistas de dados altamente qualificados.

O ***aprendizado não supervisionado*** promete aprendizado eficaz usando **dados não rotulados** e nenhuma supervisão humana. Essa é uma vantagem importante em comparação com o método *supervisionado*, pois o texto não rotulado é abundante, mas os conjuntos de dados rotulados geralmente são mais complexos e caros.

As aplicações mais populares de ***aprendizado não supervisionado*** em PLN avançado são métodos de **agrupamento (*clustering*)** e **regras de associação**.

Embora os benefícios e o nível de automação trazidos pelo *aprendizado não supervisionado* sejam grandes e tecnicamente muito intrigantes, em geral, é menos preciso e confiável em comparação com o *aprendizado supervisionado*.

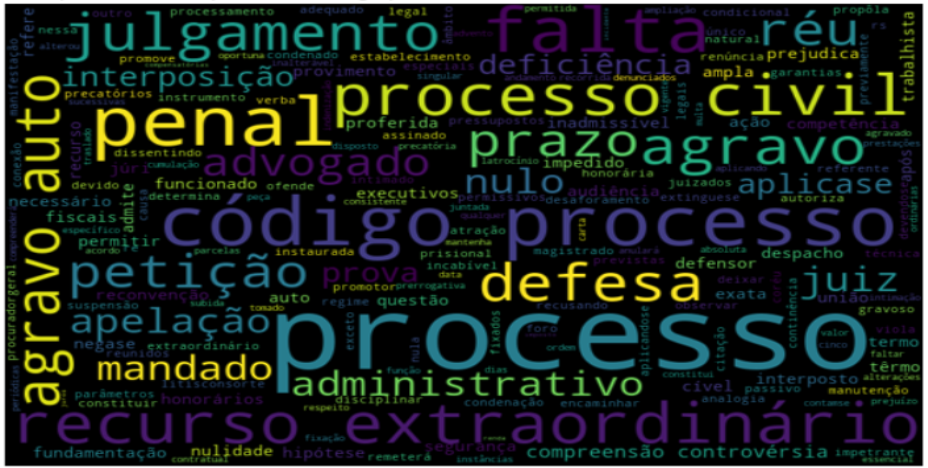
Fonte: https://aisera.com/blog/unsupervised-and-supervised-nlp-approach/

Processamento de Linguagem Natural (PLN)

Exemplo de agrupamento - aprendizagem não supervisionada – k-Means para agrupar Súmulas do STF

```
('processo', 22)('penal', 7)('código processo', 7)('falta', 6)('recurso extraordinário', 6)
('recurso extraordinário', 19)('decretolei', 6)('recurso', 6)('embargos', 5)('parágrafo', 4)
('trabalho', 19)('público', 12)('estadual', 11)('ação', 10)('competente', 8)
('militar', 23)('militares', 6)('civil', 5)('inatividade', 5)('tempo', 5)
('imposto', 41)('imposto transmissão', 14)('incide', 11)('venda', 10)('imposto circulação', 10)
('crime', 36)('penal', 13)('ação penal', 12)('pena', 10)('processo', 9)
('fiscal', 10)('crédito', 9)('inconstitucional', 8)('correção monetária', 6)('público', 5)
('anterior', 14)('vigência', 9)('embargos infringentes', 9)('julgamento', 6)('imposto', 5)
('ação', 14)('mandado segurança', 9)('sentença', 8)('salário', 7)('prescrição', 7)
('indenização', 12)('cargo', 11)('imóvel', 9)('prevista', 8)('efeito', 8)
('taxa', 36)('imposto', 11)('importação', 9)('serviço', 7)('inconstitucional', 7)
('judicial', 18)('habeas corpus', 11)('legal', 9)('ação', 8)('ato', 8)
```

('processo', 22)('penal', 7)('código processo', 7)('falta', 6)('recurso extraordinário', 6)



https://colab.research.google.com/drive/1Kl3cR2slptzNE2YLVbsJfObZYV9RkpQ3

Processamento de Linguagem Natural (PLN)

Classificação linear de texto - Naive Bayes

O algoritmo de classificação ***Naive Bayes*** (N.B.) é uma família de algoritmos probabilísticos baseados na aplicação do teorema de *Bayes* com a suposição "ingênua" de independência condicional entre cada par de um recurso.
O teorema de *Bayes* calcula a probabilidade P(c|x) onde c é a classe dos resultados possíveis e x é a instância dada que deve ser classificada, representando algumas características.
P(c|x) = P(x|c) * P(c) / P(x)
Naive Bayes prediz a *tag* de um texto. Calcula a probabilidade de cada *tag* para um determinado texto e, em seguida, produz a *tag* com a maior probabilidade.

Processamento de Linguagem Natural (PLN)

Classificação linear de texto - Naive Bayes

N.B. é simples, mas também rápido, preciso e confiável.
Parte ingênua (*naive*): assumimos que cada palavra em uma frase é independente das outras. Isso significa que não estamos mais olhando para frases inteiras, mas sim para palavras individuais. Então, para nossos propósitos, "esta foi uma festa divertida" é o mesmo que "essa festa foi divertida" e "foi uma festa divertida".
Essa suposição é muito forte, mas super útil. É o que faz esse modelo funcionar bem com poucos dados ou dados que podem ter sido rotulados incorretamente. Além disso, a redução de dimensionalidade que proporciona reduz a complexidade dos algoritmos. Por outro lado, essa suposição implica limitações de acurácia.

Processamento de Linguagem Natural (PLN)

Classificação linear de texto– Resumo de algoritmos utilizados

Para cada problema a resolver, há um algoritmo de aprendizado adequado.

- **Naive Bayes** Prós: fácil de implementar; a estimativa é rápida, exigindo apenas uma única passagem pelos dados; atribui probabilidades a rótulos previstos; controla o *overfitting* [supertreinamento] com o parâmetro de suavização. Contras: geralmente tem baixa precisão, especialmente com recursos correlacionados.
- **Perceptron** Prós: fácil de implementar; aprendizagem orientada a erros significa que a precisão é geralmente alta, especialmente após a média. Contras: não probabilístico; difícil saber quando parar de aprender; a falta de margem pode levar ao *overfitting*.

Processamento de Linguagem Natural (PLN)

Classificação linear de texto– Resumo de algoritmos utilizados

[Continuação]

Máquina de vetores de suporte (SVM) Prós: otimiza uma métrica baseada em erros, geralmente resultando em alta precisão; o *overfitting* é controlado por um parâmetro de regularização. Contras: não probabilístico.

Regressão logística Prós: orientada a erros e probabilística; o *overfitting* é controlado por um parâmetro de regularização. Contras: o aprendizado em lote requer otimização de caixa preta; a perda logística pode exagerar (*overtrain*) em exemplos rotulados corretamente.

Processamento de Linguagem Natural (PLN)

Classificação não linear de texto

O processamento de linguagem natural tem se concentrado historicamente na *classificação linear*, devido às seguintes razões:

A representação *bag-of-words* é inerentemente de alta dimensão, e o número de recursos é muitas vezes maior do que o número de instâncias de treinamento rotuladas. Isso significa que geralmente é possível encontrar um classificador linear que se ajuste perfeitamente aos dados de treinamento, ou à rotulação das instâncias de treinamento. A mudança para a ***classificação não linear*** pode, portanto, apenas aumentar o risco de *overfitting*.

Veremos, a seguir, as razões para a mudança para a predominância de classificadores não lineares.

Processamento de Linguagem Natural (PLN)

Classificação não linear de texto

Os ***classificadores não lineares*** passaram a ser a abordagem padrão para muitas tarefas. Há pelo menos 3 razões:

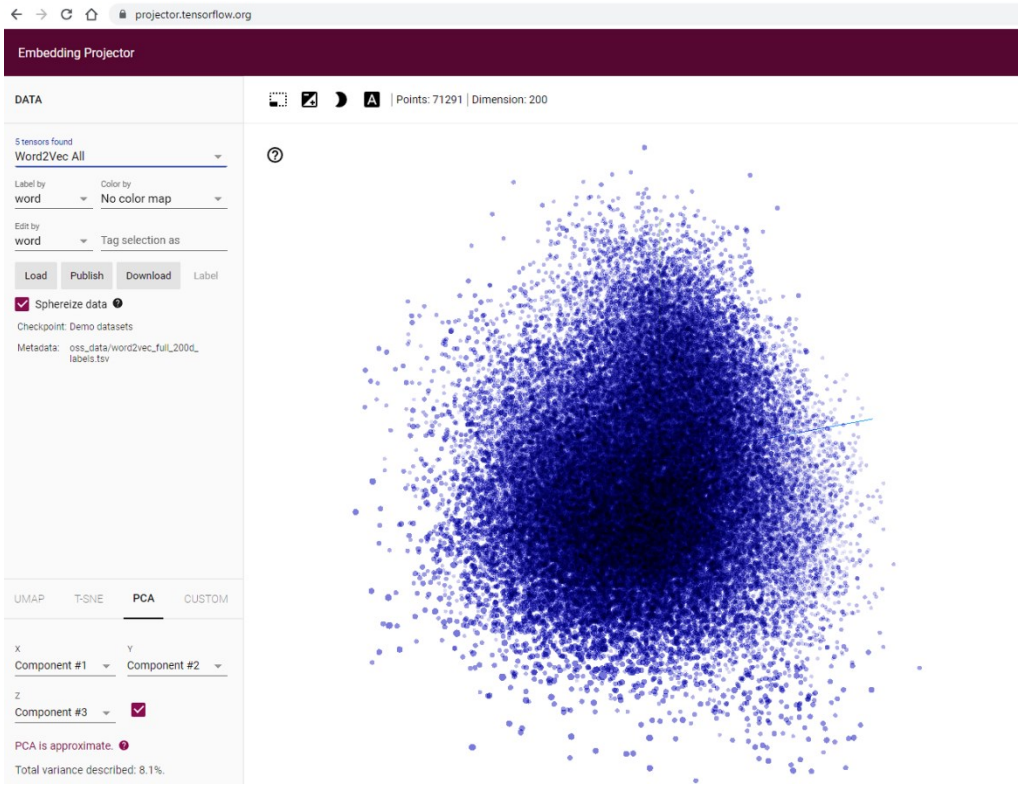
• Houve avanços rápidos no **aprendizado profundo**, uma família de métodos não lineares que aprendem funções complexas da entrada por meio de várias camadas de computação.

• O aprendizado profundo facilita a incorporação de *embeddings* de palavras, que são **representações vetoriais** densas de palavras. As incorporações de palavras podem ser aprendidas a partir de grandes quantidades de **dados não rotulados** e permitem a generalização para palavras que não aparecem nos dados de treinamento anotados.

• Houve avanços rápidos em unidades de processamento gráfico (**GPUs**), que se tornaram mais rápidas, baratas e fáceis de programar.

Processamento de Linguagem Natural (PLN)

Representação vetorial de palavras - exemplo



https://projector.tensorflow.org/

Processamento de Linguagem Natural (PLN)

BERT - Representações bidirecionais de codificadores a partir de transformadores

Aprendizado de relações contextuais

Lançado em 2018 pela equipe do Google AI, o ***BERT*** (*Bidirectional Encoder Representations from Transformers*) causou uma agitação em toda a comunidade de *Deep Learning* na época por apresentar resultados estado da arte em várias aplicações de NLP.
De uma maneira resumida, O *BERT* aplica um treinamento bidirecional em uma arquitetura de *Transformers* para treinar um modelo de linguagem. Com isso, o modelo consegue aprender **relações contextuais** entre as palavras de um texto.
Diz-se bidirecional pois a análise de relacionamento ocorre tanto em relação a antecessores quanto a sucessores de um determinado *token*.

Fonte: https://medium.com/turing-talks/ferramentas-para-processamento-de-linguagem-natural-em-portugu%C3%AAs-977c7f59c382#8eec

Processamento de Linguagem Natural (PLN)

A importância do contexto para a análise do significado das palavras - exemplos

A importância da **desambiguação do sentido das palavras** *(Word Sense Disambiguation - WSD)*.

Note que as palavras em verde dão outro sentido à palavra **banco**:

1. *Nadou* até o **banco** localizado do outro lado do rio.
2. *Caminhou* até o **banco** localizado do outro lado do rio, para *sacar* um *cheque*.

[exemplo de clusterização: https://colab.research.google.com/drive/1EDK_UnFR8ntoxcTUi1-URWtwUEVk0f3v?usp=sharing]

Contudo, neste outro exemplo, fica mais difícil a desambiguação das palavras **levante** e **cabeça**, sobretudo porque o elemento de diferenciação é o artigo, que normalmente é filtrado como *stop-word*:

1. **Levante** *a* ***cabeça***;
2. Foi *o* ***cabeça*** do **levante**.

O Google tradutor consegue fazer boa tradução desse exemplo, mas, no exemplo contido em https://colab.research.google.com/drive/1MDzolPzqnjT-TS1CD8NobLWtAPotwOxb?usp=sharing, houve resultado equivocado.

Exemplo de artigo, para conhecer um pouco mais sobre esse tema: Word Sense Disambiguation. Dhanashree Surkar, Vedika Limje, Bhavana Gopachandani – Disponível em: www.ijcseonline.org

Processamento de Linguagem Natural (PLN)

Há bastante mais assuntos sobre a matéria ...

Seria impossível esgotar a apresentação de todos os assuntos relacionados ao Processamento de Linguagem Natural em apenas uma aula. Assim, indicaremos, a seguir, algumas boas fontes de informações: artigos na Internet e obras bibliográficas

Ferramentas e artigos disponíveis na Internet

NLTK ; SpaCy ; Enelvo ; NILC embeddings ; Opinando ; BERTimbau
Os exemplos acima são listados no artigo Ferramentas para processamento de linguagem natural em português <https://medium.com/turing-talks/ferramentas-para-processamento-de-linguagem-natural-em-portugu%C3%AAs-977c7f59c382>

- Orange – aplicativo *opensource* utilizado em *machine learning* e visualização de dados - https://orangedatamining.com/
- Python toolkit SCIKIT-LEARN (https://scikit-learn.org/stable/);
- Distinções de sentido entre palavras são anotadas em WORDNET (http://wordnet.princeton.edu), um banco de dados semântico léxico para inglês.
- https://www.youtube.com/watch?v=Kc9gN_gODvQ (tutorial *Clustering with Bert Embeddings* – exemplo de como agrupar textos)

Ferramentas disponíveis (continuação)

- Lematizador online em português -> https://nlp.johnsnowlabs.com/2020/05/03/lemma_pt.html
- https://www.alura.com.br/artigos/lemmatization-vs-stemming-quando-usar-cada-uma
- http://www.clul.ul.pt/
- http://www.nilc.icmc.usp.br/lacioweb/
- http://www.corpusdoportugues.org/
- http://www.tycho.iel.unicamp.br/
- http://www.linguateca.pt/ACDC/
- http://www2.lael.pucsp.br/corpora/bp/
- https://www.youtube.com/watch?v=66seIToeguE
- https://www.youtube.com/watch?v=SZorAJ4I-sA
- https://www.youtube.com/watch?v=XowwKOAWYoQ (vídeo sobre o artigo *Attention is All You Need* - *https://arxiv.org/abs/1706.03762*)

Bibliografia

- *Li Deng and Yang Liu. 2018.* ***Deep Learning in Natural Language Processing*** *(1st. ed.). Springer Publishing Company, Incorporated.*
- Jacob Eisenstein. 2019. **Introduction to Natural Language Processing**. ISBN: 9780262042840.
- Christopher D. Manning e Hinrich Schütze. **Foundations of Statistical Natural Language Processing**. MIT Press. ISBN 0-26213360-1. 1999.
- Nitin Indurkhya e Fred J. Damerau (editores). **Handbook of Natural Language Processing**. 2ª ed. Machine Learning & Pattern Recognition Series. CRC Press.
- Christopher D. Manning; Prabhakar Raghavan;Hinrich Schütze. **An Introduction to Information Retrieval**. Cambridge University Press Cambridge, England - Online edition 2009

Bibliografia (continuação)

- Emily M. Bender. **Linguistic Fundamentals for Natural Language Processing 100 Essentials from Morphology and Syntax**. University of Washington. 2013. ISBN: 9781627050111
- Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana. **Practical Natural Language Processing**.2020. Published by O'Reilly Media, Inc. 978-1-492-05405-4
- Ramón López-Cózar Delgado, Masahiro Araki Kyoto. **Spoken, Multilingual And Multimodal Dialogue Systems Development And Assessment**. John William & Sons Ltd. 2005. ISBN-13 978-0-470-02155-2
- Rohan Chopra, Aniruddha M. Godbole, Nipun Sadvilkar, Muzaffar Bashir Shah, Sohom Ghosh, and Dwight Gunning. **The Natural Language Processing Workshop.** 2020. Packt Publishing

Links para os experimentos no Google Colab

PLN1
https://colab.research.google.com/drive/1LSMfGiMVatpLZEJ3Y6ZzzfpUOKmTAeZt?usp=sharing
PLN2
https://colab.research.google.com/drive/1whPQvP-cUJnQPReYwP0qzx_1jiR3XElM?usp=sharing
PLN3
https://colab.research.google.com/drive/1puw6jsRejkru9x3FMB9REiRB6eiyDUw5?usp=sharing
PLN4
https://colab.research.google.com/drive/1oOdnZYnUVlX3H7PwYSmXdWfpzd8dWGIN?usp=sharing
PLN5
https://colab.research.google.com/drive/1Kl3cR2slptzNE2YLVbsJfObZYV9RkpQ3?usp=sharing
PLN6
https://colab.research.google.com/drive/10kAmBlTzLzcaoBofbKi6jvUZEoeyZDvB?usp=sharing
PLN7
https://colab.research.google.com/drive/1EDK_UnFR8ntoxcTUi1-URWtwUEVk0f3v?usp=sharing
PLN8
https://colab.research.google.com/drive/1MDzolPzqnjT-TS1CD8NobLWtAPotwOxb?usp=sharing
PLN9
https://colab.research.google.com/drive/1ZQ2HhomSko3rJGKykbPTrNrN73td1zf9?usp=sharing
PLN10
https://colab.research.google.com/drive/1-3KSa2Xgh7_QB0v_IsSSaq2zw6eWef-I?usp=sharing
PLN11
https://colab.research.google.com/drive/1yU020mEdoTgiGuxIusCFNV_PZ-e0sQcs?usp=sharing