Aula 09 - Profs Thiago Cavalcanti e Erick Muzart

Banco do Brasil (Escriturário - Agente de Tecnologia) Banco de Dados - 2023 (Pós-Edital)

Autor:
Thiago Rodrigues Cavalcanti,
Erick Muzart Fonseca dos Santos,
Diego Carvalho

24 de Janeiro de 2023

Índice

#ATENÇÃO

Avisos Importantes

O curso abrange todos os níveis de conhecimento...

Esse curso foi desenvolvido para ser acessível a **alunos com diversos níveis de conhecimento diferentes.** Temos alunos mais avançados que têm conhecimento prévio ou têm facilidade com o assunto. Por outro lado, temos alunos iniciantes, que nunca tiveram contato com a matéria ou até mesmo que têm trauma dessa disciplina. A ideia aqui é tentar atingir ambos os públicos - iniciantes e avançados - da melhor maneira possível..

Por que estou enfatizando isso?

O **material completo** é composto de muitas histórias, exemplos, metáforas, piadas, memes, questões, desafios, esquemas, diagramas, imagens, entre outros. Já o **material simplificado** possui exatamente o mesmo núcleo do material completo, mas ele é menor e bem mais objetivo. *Professor, eu devo estudar por qual material?* Se você quiser se aprofundar nos assuntos ou tem dificuldade com a matéria, necessitando de um material mais passo-a-passo, utilize o material completo. Se você não quer se aprofundar nos assuntos ou tem facilidade com a matéria, necessitando de um material mais direto ao ponto, utilize o material simplificado.

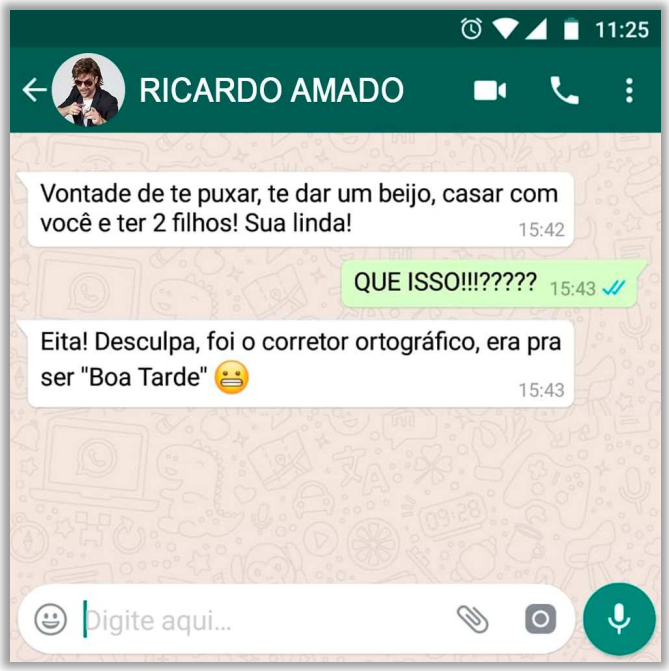
Por fim...

O curso contém diversas questões espalhadas em meio à teoria. Essas questões possuem um comentário mais simplificado porque **têm o único objetivo de apresentar ao aluno como bancas de concurso cobram o assunto previamente administrado**. A imensa maioria das questões para que o aluno avalie seus conhecimentos sobre a matéria estão dispostas ao final da aula na lista de exercícios e **possuem comentários bem mais completos, abrangentes e direcionados.**

APRESENTAÇÃO

Seus lindos, o papo agora é sobre Processamento de Linguagem Natural (PLN)! Vamos aprender como fazer uma máquina é capaz de entender e até falar a linguagem natural. *Que linguagem natural?* A linguagem natural utilizada por nós – humanos – em conversas informais, em redes sociais. Vamos falar um pouquinho de gramática, um pouquinho de estatística e bastante aprendizado de máquina. *Bora?* :)

PROFESSOR DIEGO CARVALHO - WWW.INSTAGRAM.COM/PROFESSORDIEGOCARVALHO



Galera, todos os tópicos da aula possuem Faixas de Relevância, que indicam se o assunto cai muito ou pouco em prova. *Diego, se cai pouco para que colocar em aula?* Cair pouco não significa que não cairá justamente na sua prova! A ideia aqui é: se você está com pouco tempo e precisa ver somente aquilo que cai mais, você pode filtrar pelas relevâncias média, alta e altíssima; se você tem tempo sobrando e quer ver tudo, vejam também as relevâncias baixas e baixíssimas. *Fechado?*

RELEVÂNCIA EM PROVA: BAIXÍSSIMA

RELEVÂNCIA EM PROVA: BAIXA

RELEVÂNCIA EM PROVA: MÉDIA

RELEVÂNCIA EM PROVA: ALTA

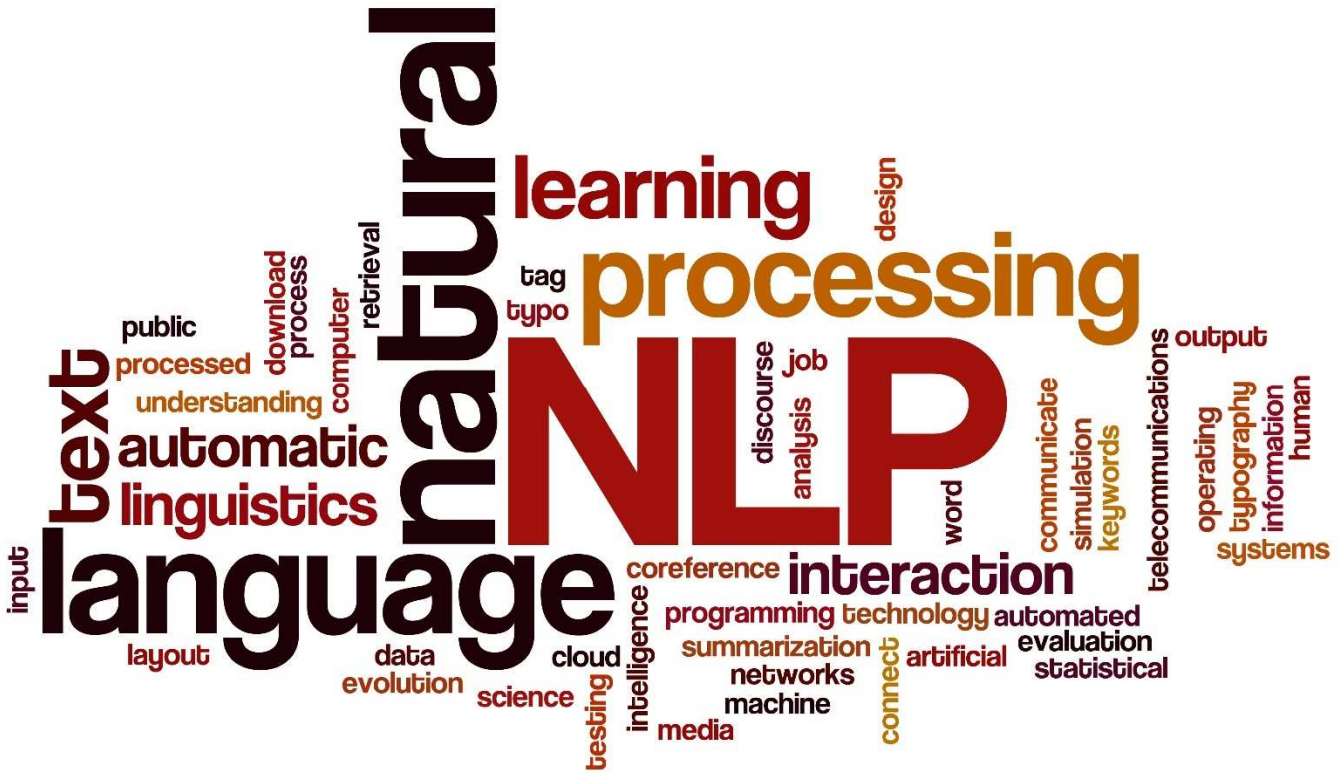
RELEVÂNCIA EM PROVA: ALTÍSSIMA

Além disso, essas faixas não são por banca – é baseado tanto na quantidade de vezes que caiu em prova independentemente da banca e também em minhas avaliações sobre cada assunto...

PROCESSAMENTO DE LINGUAGEM NATURAL

Conceitos Básicos

RELEVÂNCIA EM PROVA: MÉDIA



PROCESSAMENTO DE LINGUAGEM NATURAL (PLN)

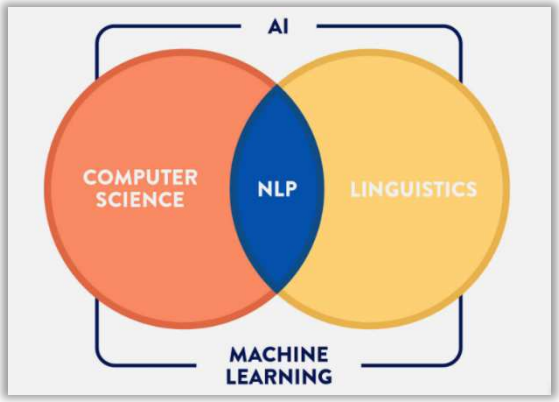
Trata-se de um ramo da inteligência artificial que ajuda os computadores a entender, interpretar e manipular a linguagem humana. O PNL permite que as máquinas leiam e entendam a linguagem humana para interpretar comandos, responder a perguntas e realizar tarefas. Ele é usado em muitas aplicações, como tradução automática, atendimento ao cliente automatizado e assistentes pessoais inteligentes.

Galera, essa aula entra em um nível de abstração bastante profundo! A leitura de um livro sobre esse assunto seria extremamente técnica e muito sacrificante para vocês. Logo, o meu papel aqui é traduzir essa complexidade de forma que vocês consigam entender. Dito isso, eu vou tentar explicar o que é o processamento de linguagem natural de três formas diferentes. **Para quem não gosta das minhas histórias e contextualizações, sugiro pular seis páginas e ir direto ao ponto...**

Quando você é criança, você aprende a falar! *Falar o quê, professor?* Em geral, você começa com "mamãe" e "papai" e com o passar do tempo você aprende mais e mais palavras, assim como entende o relacionamento entre elas. Depois de alguns anos, você vai para a escola e começa a ter

aulas de língua portuguesa. Lá, você aprende coisas como vocabulário, ortografia, gramática, morfologia, sintaxe, entre outros.

Os anos passam e você se forma na escola! Nesse momento, você provavelmente terá um domínio relativo das regras que dominam a língua portuguesa. Ocorre que, em muitas situações, nós queremos que os computadores entendam a linguagem natural. Ora, computadores são máquinas capazes de fazer cálculos muito mais rápido do que nós – **o problema é que a maneira humana de aprender a linguagem natural é muito difícil de replicar em um modelo computacional.**



Humanos são uns seres muito bons em aprender coisas – e computadores, não. Pois bem... os cientistas da computação e os especialistas em linguística se reúnem há décadas com o intuito de criar modelos computacionais que processam e entendem a linguagem natural – o nome disso é **Processamento de Linguagem Natural (PLN)**. Busca-se fazer com que o computador entenda como agrupar palavras semanticamente, transformar fala em texto, traduzir idiomas diferentes, entre outros.

Vamos para a nossa segunda explicação: desde que o primeiro computador foi inventado, a melhor maneira de se comunicar com ele tem sido usando linguagens de programação (Ex: Python, Java, C, etc). Computadores são muito melhores do que os humanos em muitas tarefas, como prever o tempo, calcular a trajetória de foguetes ou detectar e-mails de spam, e podemos usar essas linguagens para dizer aos computadores exatamente o que queremos que eles façam.

No entanto, a utilização dessas linguagens tem que ser extremamente precisa, específica e estruturada – uma vírgula errada e o computador não consegue mais processar nada ou, pior, processa de maneira incorreta. *Por que?* **Porque computadores são excelentes em processar dados estruturados, mas a maneira como duas pessoas conversam entre si por meio de sua linguagem natural não é uma maneira estruturada.** *Querem ver um exemplo?*

Se eu encontro o Professor Renato da Costa após dois anos de pandemia e falo: "*Renato, da próxima vez que você vier, traga o carro novo!*". O Renato entenderá que eu estou falando sobre vir para minha casa (em vez de outro local) e estou pedindo a ele para trazer seu carro novo (e não o carro novo de outra pessoa). *Simples, não?* Para nós, sim; para um computador, isso não é nada específico ele não compreenderá o que eu disse...

O computador não é capaz de capturar as nuances da frase! Ele não sobre que lugar eu me refiro, não entende de qual carro novo eu estou falando, não sabe de quem é o carro novo, não sabe de qual carro eu estou falando, entre outros. Na prática, um computador é muito burro – ele só é capaz de entender se eu especificar de maneira extremamente detalhada. No entanto, novas tecnologias têm surgido para resolver esse problema: a principal é o Processamento de Linguagem Natural!

Por fim, vamos à nossa terceira – e mais aprofundada – explicação: nós já sabemos que computadores entendem linguagens de programação. Para um software funcionar, são realizadas análises léxicas, análises sintáticas e análises semânticas de seu código-fonte. No entanto, essas linguagens geralmente possuem um vocabulário pequeno e seguem padrões altamente estruturados e frequentes entre linguagens (Ex: Python possui apenas 33 palavras-chave).

PASCAL	ADA	C	SHELL SCRIPT	PYTHON

`if a>0 then`
`  writeln(“Sim”)`
`else`
`  writeln(“Não”)`
	`if a>0 then`
	`  Put_Line(“Sim”);`
	`else`
	`  Put_Line(“Não”);`
	`end if;`
		`if (a>0) {`
		`  printf(“Sim”);`
		`} else {`
		`  printf(“Não”);`
		`}`
			`if [$a -gt 0]; then`
			`  echo “Sim”`
			`else`
			`  echo “Não”`
			`fi`
				`if a>0:`
				`  print “Sim”`
				`else`
				`  print “Não”`

Os códigos apresentados acima foram escritos em cinco linguagens de programação diferentes e fazem exatamente a mesma coisa – note como utilizam poucas palavras e apresentam um padrão estruturado bastante parecido entre si. Pois é... quando humanos utilizam sua linguagem natural, a história é bem diferente! Nós falamos idiomas que contêm vocabulários gigantescos, palavras com vários significados diferentes, falantes com sotaques diferentes e vários jogos de palavras.

Além disso, pessoas também cometem erros linguísticos ao escrever e falar, erram vocábulos, erram conjugações, erram a pronúncia e omitem detalhes importantes para que determinadas coisas fiquem propositalmente ambíguas. Só que – em uma conversa – podemos superar todos esses desafios de modo que tudo passe despercebido. *Como assim, Diego?* Vejamos um exemplo clássico: se eu dissesse alguma das frases a seguir...

PEGA UMA ÁGUA PARA MIM, POR FAVOR?	PEGA UMA ÁGUA PRA MIM, POR GENTILEZA?
POR FAVOR, PEGA UMA ÁGUA PRA EU?	MIM PEGA UMA ÁGUA, POR FAVOR?
POR OBSÉQUIO, VOCÊ PEGARIA UM COPO D'ÁGUA PARA MIM?	POR GENTILEZA, VOCÊ PODERIA PEGAR UM COPO CONTENDO ÁGUA?
ÁGUA, POR FAVOR?	EU GOSTARIA DE UM COPO D'ÁGUA, POR FAVOR!

Vejam quantas maneiras diferentes alguém poderia requisitar um copo com água: por meio de uma pergunta; por meio de uma afirmação; contendo erros gramaticais; contendo erros sintáticos; contendo erros morfológicos; contendo erros de conjugação; contendo erros de vocabulário; entre outros. Nós, humanos, somos sinistros... nós conseguimos pegar o sentido da frase mesmo representada de dezenas de maneiras diferentes.

O bom uso da linguagem é uma parte importante do que nos torna humanos e, por esta razão, o desejo dos pesquisadores de que os computadores entendam (e falem) a nossa linguagem natural existe há décadas. **Bem, foi isso que levou à criação do Processamento de Linguagem Natural (PLN): um campo interdisciplinar da inteligência artificial que combina ciência da computação e estudos de linguística.**

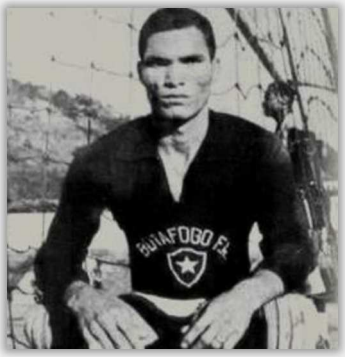
Conforme vimos acima, há um número essencialmente infinito de maneiras de organizar as palavras em uma frase e nós não podemos simplesmente dar aos computadores um dicionário de todas as frases possíveis para ajudá-los a compreender sobre o que os humanos estão falando.

Logo, um problema inicial e fundamental do processamento de linguagem natural era desconstruir frases em pedaços pequenos de forma que elas pudessem ser processadas mais facilmente.

Vamos voltar à escola: nós aprendemos que a língua portuguesa possui dez tipos ou classes de palavras: substantivos, pronomes, artigos, numerais, verbos, adjetivos, advérbios, preposições, conjunções e interjeições – estes são também conhecidos como classes gramaticais. Para cada um desses, existem todos os tipos de subcategorias (Ex: adjetivo simples, composto, primitivo ou derivado; substantivos concretos e abstratos; etc).

ARTIGO	SUBSTANTIVO? VERBO?	VERBO	PREPOSIÇÃO	ARTIGO	SUBSTANTIVO
A	MANGA	VEIO	PARA	O	BRASIL

Ora, saber a classe de uma palavra é extremamente útil! Por outro lado – infelizmente – existem diversas palavras que possuem vários significados diferentes. Vejamos a palavra **manga**:

MUNICÍPIO DO ESTADO DE MINAS GERAIS	JOGADOR DA SELEÇÃO BRASILEIRA DE 1966	TIPO DE FRUTA
 Localização de Manga em Minas Gerais		
PARTE DA ROUPA QUE COBRE O BRAÇO	**GÊNERO DE TRAÇA DA FAMÍLIA ARCTIIDAE**	**FLEXÃO DO VERBO MANGAR**
		

Note que essa palavra pode ser um substantivo comum, um substantivo próprio ou a flexão de um verbo. Logo, saber a classe da palavra ajuda, mas não é suficiente para resolver a ambiguidade que existe para um computador – é necessário também saber um pouco de gramática. Dessa forma, pesquisadores implementaram um conjunto de regras de estruturas de frases que representam a gramática de um idioma. *Como é, Diego?* Vejamos...

Em português, nós sabemos que – em regra – uma oração é composta de um sujeito e um predicado! No entanto, o predicado pode ser composto de um verbo e um objeto; uma oração pode não ter um sujeito; o sujeito pode ser composto de um artigo e um substantivo; o verbo pode seguir ou preceder um advérbio; o objeto pode ser seguido de um adjetivo. Enfim... é possível documentar as estruturas de orações como um conjunto de regras que computadores devem analisar.

Por meio da utilização dessas regras, é fácil construir uma Árvore de Análise Sintática (*Parse Tree*). *O que é isso, Diego?* É um algoritmo que permite marcar cada palavra com sua provável classe gramatical, assim como pode revelar como a frase é construída. Na frase inicial, temos a estrutura "*A manga veio (...)*". Ora, é mais provável que essa estrutura seja [artigo][substantivo][verbo] do que [artigo][verbo][verbo]. Logo, **manga** provavelmente é um substantivo...

A divisão e classificação de partes de uma frase ajuda os computadores a acessarem, processarem e responderem com mais facilidade às informações – é o famoso *dividir para conquistar*. Processos semelhantes acontecem toda vez que você faz uma pesquisa por voz utilizando algum assistente virtual (Ex: Apple Siri, Google Assistant, Amazon Alexa ou Microsoft Cortana). Você pode perguntar em seu smartphone agora: "*Onde é a pizzaria mais próxima?*".

O assistente virtual reconhecerá que esta é uma questão de localização por conta do "*onde*"; ele sabe que você quer o substantivo "*pizzaria*"; e a dimensão com a qual você se preocupa é "*mais próxima*". O mesmo processo se aplica à pergunta "*Qual é a maior girafa*?" ou "*Quem canta a música Thriller*?". Ora, o computador trata a linguagem natural como um lego que você pode desmontar e remontar cada peça.

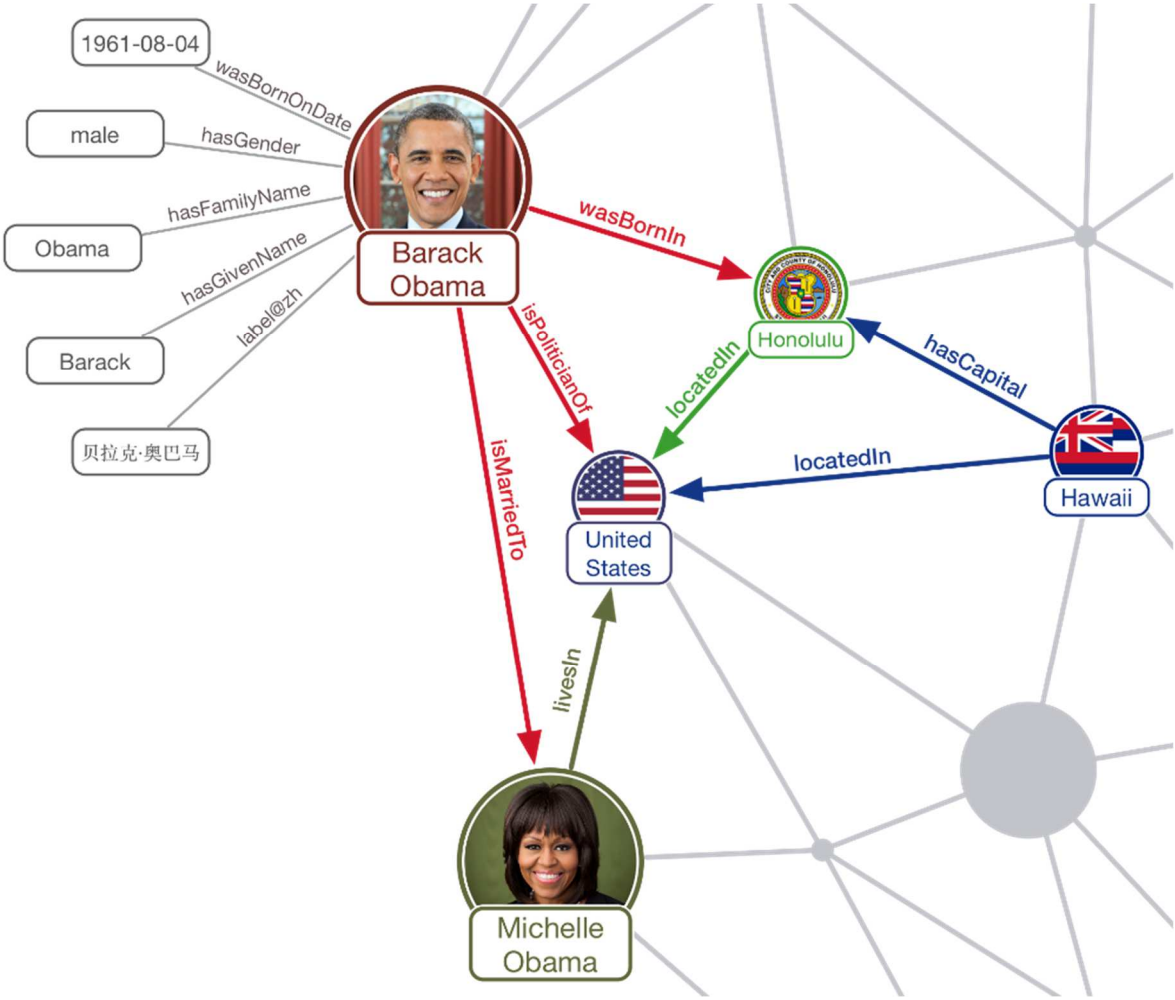
Lembrando que isso não vale apenas para perguntas! Assistentes virtuais entendem comandos como "*Eu quero um alarme para 14 horas*". Eu sei que muitas vezes o assistente virtual não entende o que falamos, mas a tecnologia ainda tem muito a avançar. Se você fala algo de maneira um pouco mais sofisticada, ele ainda não consegue analisar corretamente a frase e capturar sua verdadeira intenção ou sentido.

Agora olha que interessante: nós estamos nos focando muito em como computadores podem entender linguagem natural, mas existe também um foco em como computadores podem falar em linguagem natural. *Como é, Diego?* Pessoal, as regras de estrutura de uma frase podem ser utilizadas por computadores tanto para entender o que um humano está falando quanto para gerar um texto em linguagem natural para falar com um humano.

Isso funciona particularmente bem quando os dados são armazenados em uma rede de informações semânticas, onde entidades estão ligadas umas às outras em relacionamentos significativos, fornecendo todos os ingredientes você precisa criar frases informativas. *Professor, eu não entendi bulhufas!* Vejam só: eu acabei de acionar a Alexa em meu smartphone e perguntar: "*De que morreu Belchior*?". A assistente virtual pesquisou em uma de suas fontes, juntou blocos de

acordo com as regras de estrutura de frases e me respondeu: "*Belchior faleceu em 30 de abril de 2017 aos 70 anos de idade devido a aneurisma da aorta*".

Logo, aquelas regras que nós vimos servem tanto para entender a fala humana quanto para falar como um humano. *E como ele faz isso?* No Google, existe um recurso chamado Grafo do Conhecimento. Trata-se de um banco de dados com bilhões de fatos sobre pessoas, lugares e coisas. O seu objetivo é que os sistemas descubram e mostrem informações factuais publicamente conhecidas quando forem consideradas úteis. Vamos ver um exemplo bacana com o ex-presidente Barack Obama:



Vejam acima que é possível saber que Barack Obama é um político americano de Hawaii, do sexo masculino, nascido em 04 de agosto de 1961, casado com Michelle Obama, entre outros. *Entenderam agora o que eu quis dizer?* Se uma máquina sabe como montar frases de acordo com as regras gramaticais que nós implementamos e sabe como informações se conectam por meio de um grafo de conhecimento, ela é capaz de responder perguntas com a linguagem natural humana.

VÍDEO INTERESSANTE SOBRE KNOWLEDGE GRAPH:

HTTPS://WWW.YOUTUBE.COM/WATCH?V=MMQL6VGVX-C

Aliás, é agora que chegamos nos famosos chatbots! Por meio dos processos de análise e geração de texto em linguagem natural, é possível implementar robôs que conversam com humanos. Eu sei, eu sei que eles muitas vezes não funcionam muito bem e chegam a ser irritantes. Você liga na sua operadora de telefonia para resolver um problema e um chatbot te responde de maneira insatisfatória. É verdade, mas calma... a tecnologia está avançando!

Os primeiros chatbots eram baseados principalmente em regras em que especialistas codificavam para mapear o que um usuário poderia dizer e como um programa deveria responder. É claro que era difícil oferecer respostas sofisticadas! Em 1964, Joseph Weizenbaum criou um software chamado Eliza – considerado como primeiro chatbot do mundo – bastante rudimentar para simular a conversa entre um paciente e sua psicóloga/terapeuta (Eliza). Vejamos um exemplo:

```
- Eliza: Por favor, diga o seu problema!
- Usuário: Eu estou com receio de que eu não sou um bom artista.
- Eliza: Você veio até mim porque você está com medo de não ser um bom artista?
- Usuário: Sim, eu estou com medo de ser um artista ruim sem saber se realmente sou.
- Eliza: Você parece positivo.
- Usuário: Eu me sinto muito negativo.
- Eliza: Você disse que se sente muito negativo?
- Usuário: Sim, como se eu não fosse um bom artista e os outros riem de mim por eu ser ruim.
- Eliza: Você se sente como se não fosse um bom artista?
```

Veja que é tudo muito simples: o computador basicamente utiliza palavras-chave que o usuário escreve ou fala, e faz uma pergunta as utilizando. Se o usuário diz: "*Estou preocupado com minha esposa*", o computador teria uma regra que responderia com algo como "*Fale mais sobre sua esposa*". Surpreendentemente, esse programa simples funcionou bem o suficiente para que muitas pessoas estivessem dispostas a falar com o programa como fariam com um terapeuta humano.

Infelizmente, acabou sendo muito trabalhoso usar essa abordagem para construir programas que fazem coisas mais sofisticadas com a linguagem. Atualmente, a abordagem mais utilizada é o Machine Learning (ML), que é capaz de fazer com que os computadores entendam melhor a linguagem humana com todas as suas peculiaridades, ambiguidades, nuances, entre outros. *Como isso é feito, Diego?*

Por meio gigabytes e gigabytes de conversas reais entre humanos, que são utilizadas para treinar os *chatbots*. Hoje, a tecnologia está encontrando uso em aplicativos de atendimento ao cliente, onde já existe uma infinidade de exemplos de conversas que permitem que o algoritmo aprenda. Aliás, toda vez que utilizamos ferramentas que utilizam aprendizado de máquina, criamos um feedback positivo em que acabamos treinando a própria ferramenta.

Toda vez que você fala com um assistente virtual, você fornece dados de entrada para treinar cada sistema. Isso permite uma resposta mais correta e precisa, o que acaba fazendo com que as pessoas usem cada vez mais essas ferramentas, o que novamente permite melhorar ainda mais a precisão e corretude, e assim por diante em um loop infinito que futuramente as máquinas falarão tão próximo da linguagem natural que não poderemos diferenciar mais máquina e humano.

Muitos preveem que as tecnologias de fala se tornarão uma forma de interação tão comum quanto telas, teclados, *trackpads* e outros dispositivos físicos de entrada e saída que usamos hoje. Bem, pessoal... é isso! Eu gosto de contar histórias, dar exemplos e inserir uma contextualização para que vocês entendam de maneira mais amigável. Agora vamos ver algumas definições técnicas de processamento de linguagem natural:

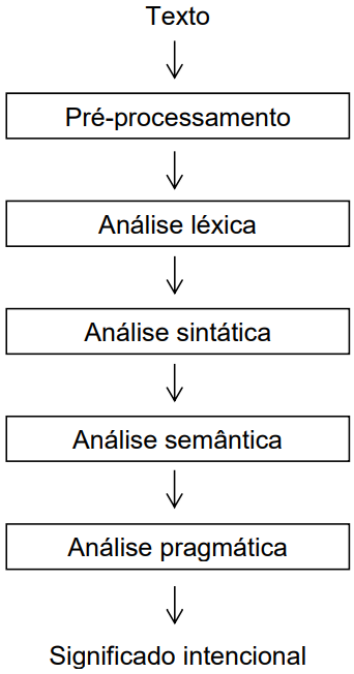
DEFINIÇÕES DE PROCESSAMENTO DE LINGUAGEM NATURAL
Trata-se da tecnologia que envolve a habilidade de transformar texto ou áudio em informações estruturadas e codificadas, baseado em uma ontologia adequada.
Trata-se da habilidade de um programa de computador de compreender a linguagem humana escrita e falada.
Trata-se da habilidade construir um software capaz de analisar, compreender e gerar linguagens humanas naturalmente, permitindo a comunicação com um computador como se fosse um humano.
Trata-se do campo da Inteligência Artificial que permite aos computadores analisar e compreender a linguagem humana, escrita e falada.
Trata-se da capacidade de construir software que gere e compreenda linguagens naturais para que um usuário possa ter conversas naturais com um computador em vez de por meio de programação.
Trata-se do ramo da inteligência artificial que ajuda os computadores a entender, interpretar e manipular a linguagem humana.
Trata-se da manipulação automática da linguagem natural, como fala e texto por software.
Trata-se de uma subárea da ciência da computação, inteligência artificial e da linguística que estuda os problemas da geração e compreensão automática de linguagens humanas naturais.

O Processamento de Linguagem Natural (PLN) está ligado a três aspectos da comunicação em língua natural, quais sejam:

FONOLOGIA	Está relacionada ao reconhecimento de sons que compõem as palavras.
MORFOLOGIA	Reconhece as palavras em termos das unidades primitivas que a compõem.
SINTAXE	Define a estrutura de uma frase, com base na forma como as palavras se relacionam.
SEMÂNTICA	Associa significado a uma estrutura sintática, em termos dos significados das palavras que a compõem.
PRAGMÁTICA	Verifica se o significado associado a uma estrutura sintática é realmente o significado mais apropriado no contexto considerado.

PROCESSAMENTO DE LINGUAGEM NATURAL		
SOM	ESTRUTURA	SIGNIFICADO
FONOLOGIA	MORFOLOGIA + SINTAXE	SEMÂNTICA + PRAGMÁTICA

Robert Dale (*Handbook of Natural Language Processing*, 2010) afirma que o trabalho do processamento de linguagem natural tende a ver o processo de análise da linguagem como uma decomposição em estágios, iniciando o exame na superfície do texto e aumentando em cada passo a profundidade da análise. *Como assim, Diego?* Vejam na imagem a seguir uma sugestão de estágios de análise em processamento de linguagem natural...

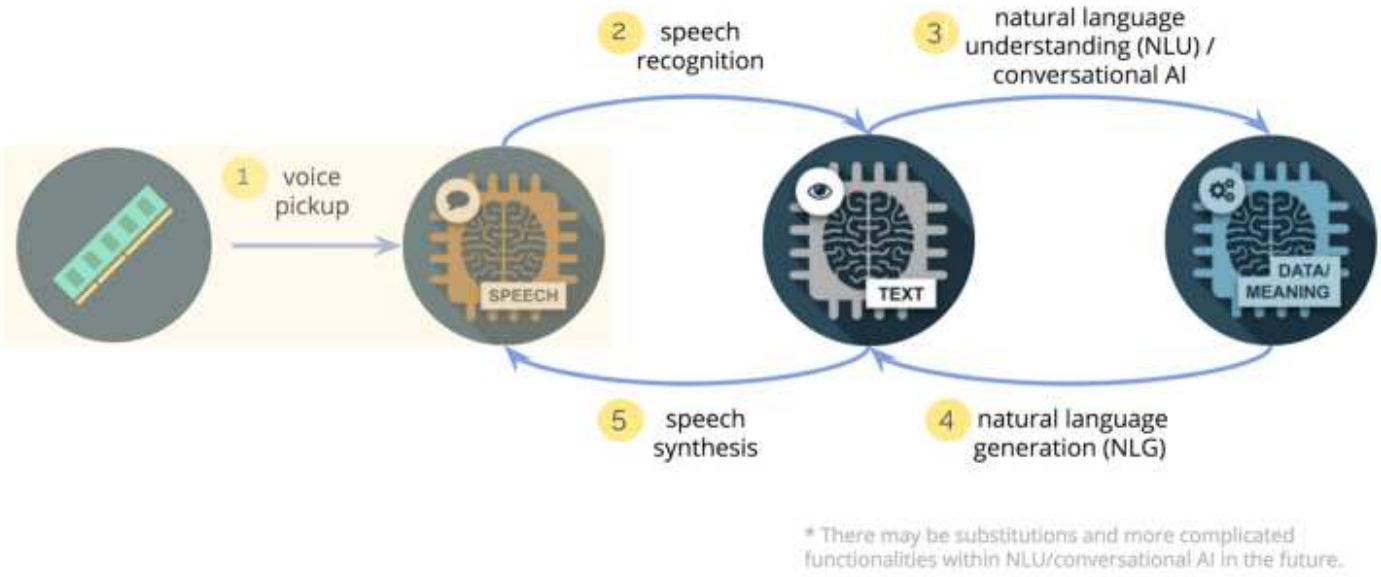


PRÉ-PROCESSAMENTO	Trata-se do estudo de estruturas e formação de palavras, com foco na análise dos componentes individuais das palavras. Nesse contexto, trata-se basicamente da realização da tarefa de tokenização (veremos à frente outro contexto).
ANÁLISE LÉXICA	Busca estudar a morfologia das palavras e recuperar informação que será útil em níveis mais profundos de análise. Para tal, realiza uma decomposição morfológica para identificar classes gramaticais de cada um dos tokens selecionados na atividade anterior.
ANÁLISE SINTÁTICA	A análise sintática é aquela que se preocupa com a estrutura das sentenças em uma gramática formal. Ela permite a extração de frases que transmitem mais significado do que apenas as palavras individuais por si só.
ANÁLISE SEMÂNTICA	A análise semântica trata do significado da sentença.
ANÁLISE PRAGMÁTICA	O componente pragmático, por fim, procura incluir o contexto à análise linguística, a fim de permitir a geração de um significado.

Principais Aplicações

RELEVÂNCIA EM PROVA: BAIXA

Vamos ver agora as principais aplicações do processamento de linguagem natural. Para isso, vamos ver as etapas de um processo de interação linguística entre humanos e máquinas:



ETAPA	DESCRIÇÃO
CAPTAÇÃO DA VOZ (VOICE PICKUP)	Trata-se da tecnologia que utiliza um microfone para detectar ondas sonoras e convertê-las em sinais elétricos. Essa tecnologia é usada em muitos dispositivos, como telefones, computadores e sistemas de reconhecimento de voz. O microfone capta as ondas sonoras da voz do usuário e as converte em sinais elétricos, que são então processados pelo dispositivo para determinar o que o usuário está dizendo.
RECONHECIMENTO DE FALA (SPEECH RECOGNITION)	Trata-se da tecnologia que permite que um dispositivo reconheça e responda a comandos falados. Essa tecnologia pode ser utilizada para controlar um dispositivo ou aplicativo, transcrever áudio em texto ou entender comandos de linguagem natural. Em outras palavras, podemos dizer que se trata da transcrição da fala no texto correspondente ao que foi dito por um humano. Hoje em dia, essa tecnologia está excepcional! Quem costuma utilizar ferramentas de transcrição – como jornalistas – sabe que a taxa de erros é baixíssima.
ENTENDIMENTO DE LINGUAGEM NATURAL (NATURAL LANGUAGE UNDERSTANDING)	Trata-se da tecnologia que se concentra em permitir que os computadores entendam a fala humana e a linguagem natural. Envolve o desenvolvimento de algoritmos e modelos capazes de interpretar e processar a linguagem falada e extrair informações relevantes do texto em linguagem natural, além de permitir que os computadores entendam o significado por trás das palavras faladas – podendo ser usado para tarefas como atendimento automatizado ao cliente (*os famosos chatbots*).
GERAÇÃO DE LINGUAGEM NATURAL (NATURAL LANGUAGE GENERATION)	Trata-se da tecnologia que permite que os computadores gerem automaticamente uma linguagem natural em texto a partir de dados estruturados. Ele é usado em uma variedade de aplicações, incluindo *chatbots* de atendimento ao cliente, geração automatizada de artigos de notícias e assistentes virtuais. Pode ser usado para gerar resumos, relatórios e

	outras saídas de texto de fontes de dados estruturadas, como bancos de dados, planilhas e documentos XML.
SÍNTESE DE FALA (SPEECH SYNTHESIS)	Trata-se da tecnologia de produção artificial da fala humana. Por meio da síntese de fala, os computadores são capazes de gerar fala semelhante à humana usando a tecnologia Text-To-Speech (TTS). Essa tecnologia é utilizada em muitos aplicativos, como produtos habilitados para fala, aplicativos de conversão de texto em fala e assistentes virtuais. Hoje em dia, já existem sintetizadores de vozes digitais quase impossíveis de serem identificados como produzidos por uma máquina.

Quem aí já interagiu com uma Alexa? Trata-se do assistente virtual mais moderno atualmente! Quando você faz uma pergunta para a Alexa, ocorre uma captação da sua voz (Etapa 1), que é transcrita da voz para o texto (Etapa 2), em seguida é interpretada em busca de seu sentido (Etapa 4) de forma que ela consiga gerar um texto de resposta (Etapa 5) que possa ser falado por um sintetizador de voz (Etapa 6).

Dito isso, vamos ver agora as principais aplicações de processamento de linguagem natural (algumas aplicações serão mais detalhadas em tópicos específicos).

Classificação de Textos

Trata-se do processo de atribuição de categorias predefinidas ao texto de acordo com seu conteúdo. É uma forma de mineração de texto que usa algoritmos de aprendizado de máquina para ordenar e classificar documentos, texto, etc. Esse processo pode ser usado para rotular automaticamente grandes quantidades de dados, como artigos de notícias, postagens de blog, e-mails, etc e para organizar dados em categorias (Ex: tópicos, sentimento, autor e assim por diante).

Análise de Sentimentos

Trata-se de um subtipo de Classificação de Texto que busca determinar computacionalmente a atitude ou opinião de um texto em linguagem natural. Essa tecnologia pode ser utilizada para determinar se um texto é positivo, negativo ou neutro. É comumente usada em marketing para identificar o sentimento do cliente sobre produtos e serviços e também pode ser usado para análise de mídia social a fim de identificar tendências e opiniões sobre tópicos de interesse.

Identificação de Idioma

Trata-se do processo de detecção e reconhecimento de idiomas em um texto de linguagem natural, como sentenças e frases. Isso pode ser feito usando ferramentas automatizadas, a fim de identificar e interpretar expressões idiomáticas por meio de técnicas de aprendizado de máquina. A identificação de idiomas pode ajudar a melhorar a precisão dos sistemas de processamento de linguagem natural e a compreensão de expressões idiomáticas.

Reconhecimento de Fala (Transcrição)

No contexto de processamento de linguagem natural, o reconhecimento de fala é a capacidade de uma máquina ou programa de reconhecer e responder a palavras e frases faladas. Trata-se basicamente de uma transcrição de sinais digitais de voz em texto escrito. Essa tecnologia pode ser usada em uma variedade de aplicativos, como interfaces de usuário por voz, pesquisa por voz, comandos de voz e transcrição automatizada de áudio.

Análise Gramatical

Trata-se do processo de análise da estrutura de uma frase para dividi-la em seus componentes gramaticais (ou seja, substantivos, verbos, adjetivos, advérbios, etc). Essa análise é usada para identificar o significado de uma frase ou para entender como ela é estruturada. A análise gramatical é uma parte fundamental do processamento de linguagem natural e é usada em muitas aplicações, como tradução automática, resposta a perguntas e resumo de texto.

Extração de Informação

Trata-se do processo de identificar e extrair informações desejadas a partir de documentos, sejam estes estruturados ou não, armazenando-as em um formato mais apropriado para consultas semânticas futuras. Ela também pode identificar relacionamentos entre entidades e determinar sentimentos. Por exemplo: é possível analisar diversos documentos em formato PDF em busca de informações específicas sem a necessidade de utilização de busca por palavras-chave.

Análise Semântica

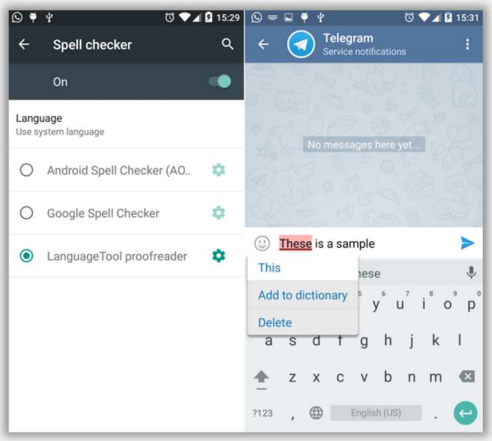
Trata-se do processo de compreensão do significado de um trecho de texto, examinando as relações entre palavras, frases e sentenças. Envolve analisar o contexto do texto e usar o conhecimento do idioma para entender seu significado. Em geral, busca identificar textos similares em um conjunto de documentos, analisando papeis e relacionamentos entre objetos a fim de determinar seus sentidos.

Modelagem de Conhecimento

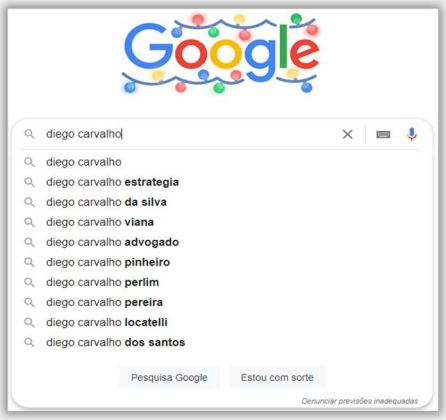
Trata-se do processo de criar modelos semânticos que representam a estrutura e o significado de um conjunto específico de conhecimento. Um modelo de conhecimento normalmente inclui um conjunto de entidades e relacionamentos entre elas que são usados para representar fatos e regras relacionadas a um determinado domínio de conhecimento. Pode ser usada também para permitir aplicações que respondem perguntas, sumarizam textos e reconhecem intenções.

Por fim, outras aplicações de processamento de linguagem natural podem ser resumidas na tabela seguinte. Vejamos...

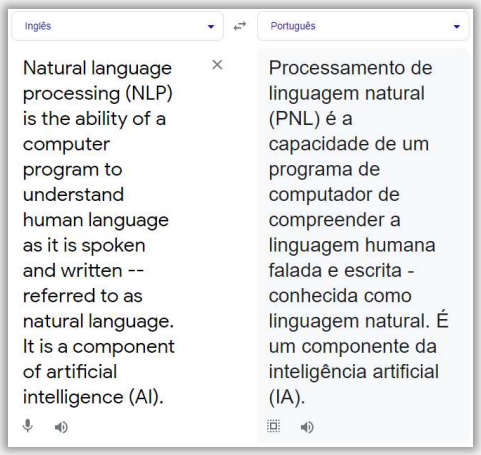
CORRETORES ORTOGRÁFICOS



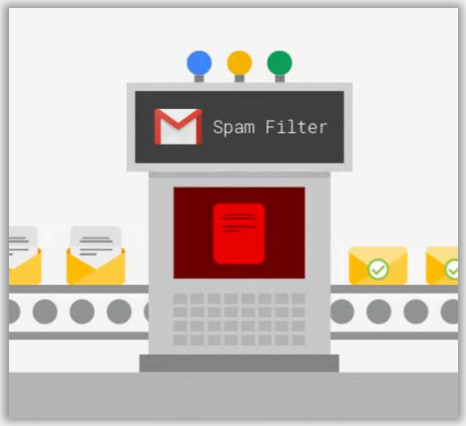
FUNÇÃO AUTO-COMPLETAR



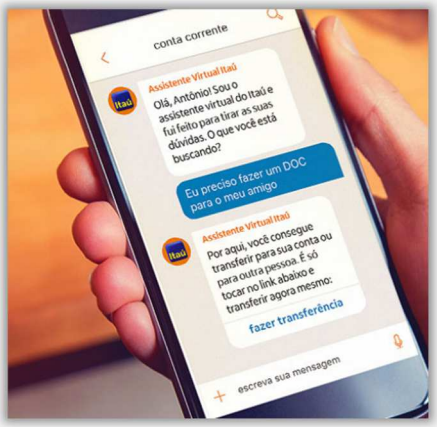
FERRAMENTAS DE TRADUÇÃO



FILTRO DE SPAM



BOTS DE MENSAGENS



ASSISTENTE VIRTUAL

Pré-Processamento

RELEVÂNCIA EM PROVA: BAIXA

PRÉ-PROCESSAMENTO EM PLN

Trata-se de uma variedade de técnicas usadas para preparar o texto para análise posterior. Isso pode envolver a transformação do texto em uma forma mais receptiva à análise, normalizando o texto, limpando o texto e/ou extraindo recursos do texto. Exemplos de técnicas de pré-processamento incluem tokenização, remoção de stopword, lematização, entre outros.

Para representar o texto de forma adequada e modelar uma linguagem que possibilite o melhor entendimento da máquina, são necessários alguns pré-processamentos que abstraem e estruturam a linguagem, deixando apenas o que é informação relevante. *O que você quer dizer com abstração?* Abstração é a remoção de detalhes, mantendo apenas aquilo que é essencial! O pré-processamento abstrai a linguagem para o processamento computacional.

Diego, não há problema em remover informações? Não, porque frequentemente essas informações não são relevantes ou são redundantes para o processo de classificação do texto. Dessa forma, o objetivo do pré-processamento é extrair de textos uma representação estruturada e manipulável que identifique o subconjunto mais significativo de informações, isto é, obter uma representação com qualidade melhor que a qualidade inicial.

O pré-processamento envolve intrinsecamente o conceito de normalização, isto é, o processo de remover ruídos e transformar um texto para um formato normal de forma que uma máquina possa detectar padrões. *Como é, Diego?* Galera, textos podem conter diversos ruídos, isto é, qualquer elemento que dificulte a compreensão por computadores. Logo, ela vai tratar dos símbolos, numerais, pontuações, caracteres especiais, contrações, erros ortográficos, entre outros.

Por que, professor? Basicamente porque eles – em geral – não agregam nenhuma informação relevante em termos de semântica para facilitar o entendimento da máquina. Além disso, a normalização busca transformar um texto para um formato normal. *Como é isso, Diego?* Pessoal, é possível dizer uma mesma coisa por meio de formatos diferentes. A normalização busca reduzir a aleatoriedade, levando o texto para um formato mais bem padronizado.

Vocês se lembram que computadores gostam de dados estruturados? Pois é, trata-se de uma maneira de chegar mais próximo desses dados estruturados e padronizados. Isso nos ajuda a reduzir a quantidade de informações diferentes com as quais o computador precisa lidar e, portanto, melhora a eficiência – quanto menor o número de variáveis, mais fácil de lidar. Bem... ao fazer a normalização de texto, devemos saber exatamente o que queremos normalizar e por quê.

Basicamente existem duas coisas que queremos normalizar: estrutura e vocabulário. Para tal, podemos utilizar diversas técnicas de pré-processamento:

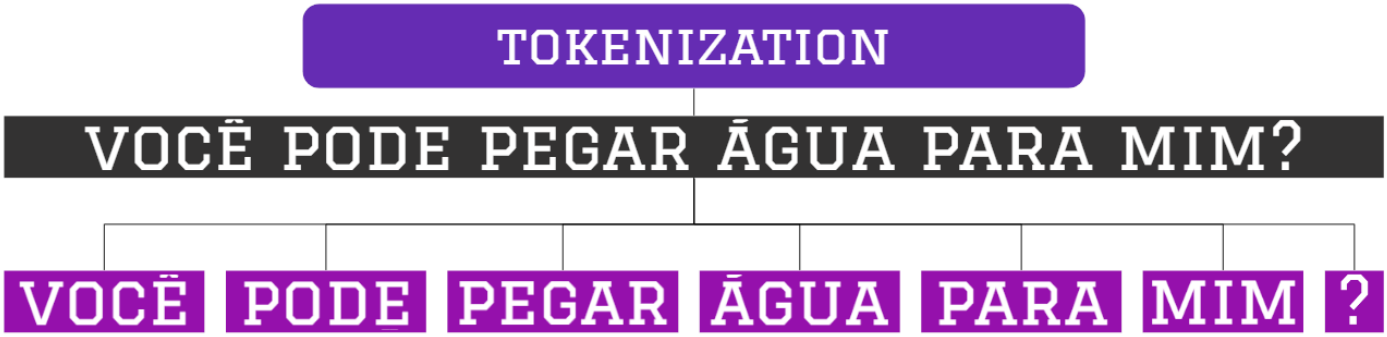
EXEMPLOS DE TÉCNICAS DE NORMALIZAÇÃO OU LIMPEZA DE DADOS
Remover espaços em branco e pontuação duplicados.
Remover acentuação gráfica, visto que isso ajuda a reduzir erros relacionados ao tipo de codificação de caracteres.
Transformar maiúsculas em minúsculas (exceto quando queremos extrair informações como nomes e locais).
Remover ou substituir caracteres especiais (Ex: &$#@) e emojis (Ex: 😊).
Remover contrações (Ex: caixa d'água).
Transformar numerais das palavras em números (Ex: vinte e três → 23).
Substituir valores pelo seu tipo (Ex: R$50 → Dinheiro; 100Kg → Peso).
Normalizar siglas (Ex: RJ → Rio de Janeiro) e abreviações/vocabulários informais (Ex: pfv → por favor).
Normalizar formatos de data, números de CPF ou outros dados que tenham um formato padrão definido.
Corrigir a ortografia de palavras incorretas.
Remover variações de gênero, tempo, grau e número.
Substituir palavras raras por sinônimos mais comuns.
Remover tags HTML, CSS, JavaScript, etc – além de URLs.
Padronizar de palavras com caracteres minúsculos.

Lembre-se de que não existe uma lista exaustiva de tarefas de normalização que funcionem para todas as situações – tudo dependerá do contexto. Veremos a seguir algumas com mais detalhes...

Tokenization

A tokenização, também conhecida como segmentação de palavras, é responsável por quebrar a sequência de caracteres em um texto, localizando o limite de cada palavra, isto é, os pontos onde uma palavra termina e outra começa. Para fins de linguística computacional, as palavras assim identificadas são frequentemente chamadas de tokens[1]. Para separar cada token, geralmente são utilizadas quebras de linhas, espaços em branco ou outros delimitadores.

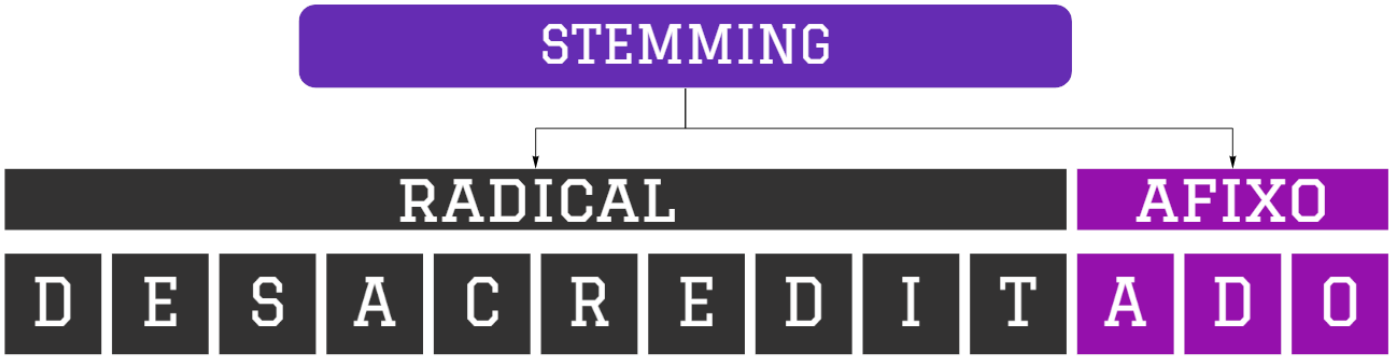
No exemplo, o separador foi o espaço! Existem idiomas que não realizam separações por espaços ou pontuações, logo requerem informações léxicas e morfológicas adicionais (Ex: tailandês).



[1] Na verdade, tokens geralmente são palavras, mas podem também ser frases ou parágrafos.

Stemming

A stemização (do inglês, *stemming*) é o processo que basicamente consiste em reduzir uma palavra ao seu radical – removidos seus afixos (prefixos, infixos e sufixos). Por exemplo: as palavras "***menin****as*", "***menin****os*", "***menin****a*", "***menin****o*", "***menin****ada*", "***menin****inhos*", "***menin****inhas*", "***menin****ão*" – todas possuem o mesmo radical[2]. Note que isso reduz o ruído, aleatoriedade ou variedade de informação a ser processada pela máquina.

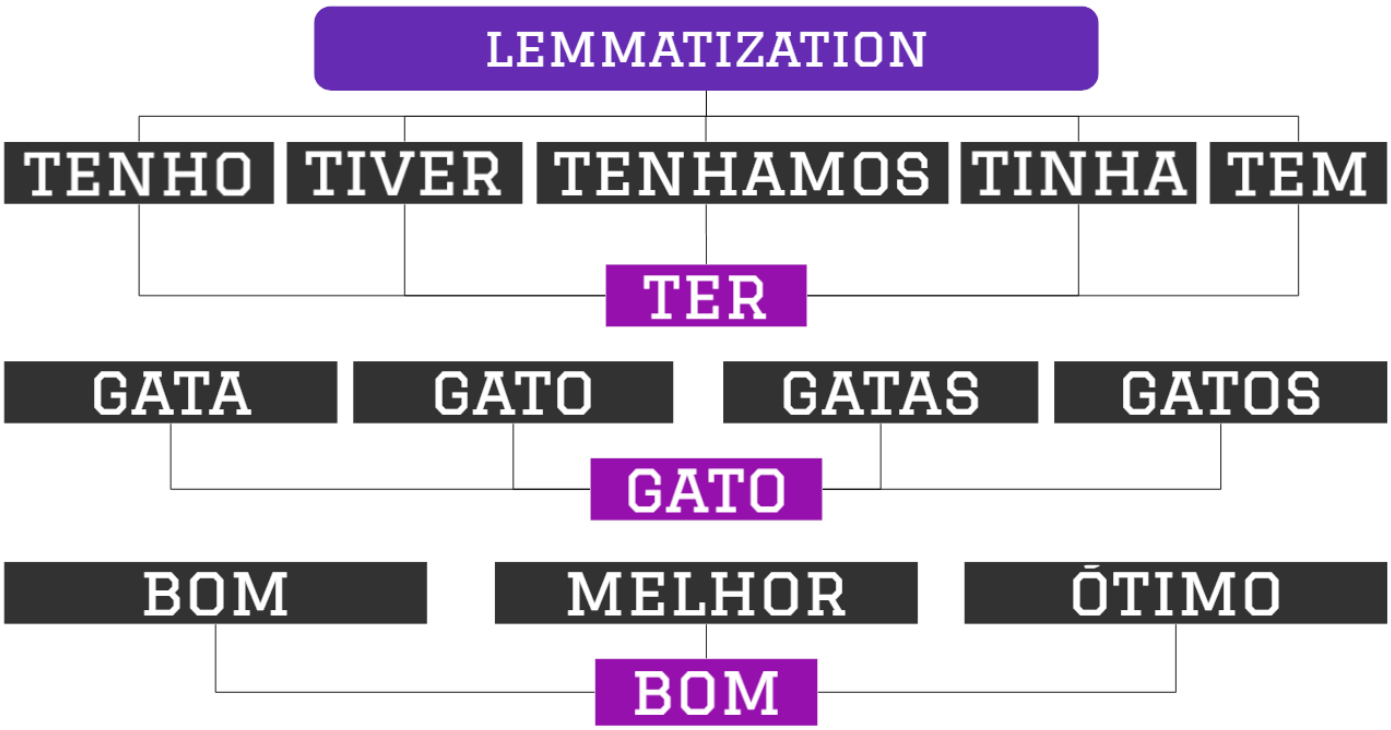


Lemmatization

A lematização (do inglês, *lemmatization*) é o processo de reduzir uma palavra ao seu lema, isto é, à sua forma base, primitiva, nuclear ou canônica – além de agrupar diferentes formas de uma mesma palavra ou seus sinônimos. *Como assim, professor?* Em geral, verbos ficam no infinitivo (Ex: "*tenho*", "*tiver*", "*tenhamos*", "*tinha*", "*tem*" → "*ter*") e substantivos/adjetivos ficam no masculino singular (Ex: "*gato*", "*gata*", "*gatos*", "*gatas*" → "*gato*").

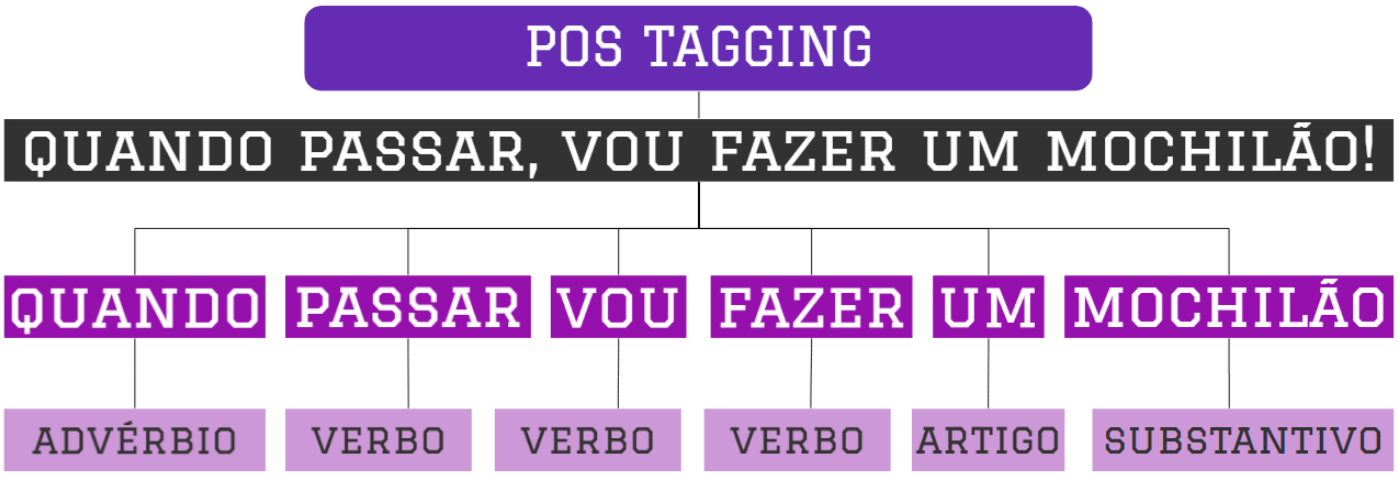
Note que a lematização aumenta a abstração reduzindo o vocabulário ao retirar inflexões das palavras, mas não é só isso: ela também é responsável por agrupar diferentes formas de uma mesma palavra ou seus sinônimos, logo "*bom*", "*melhor*" e "*ótimo*" poderiam ser agrupadas pelo mesmo lema: "*bom*". Isso ajuda a padronizar a palavra para um significado comum e estruturar um texto para o processamento de linguagem natural.

[2] Note que o radical ≠ raiz: o primeiro é a parte fixa de um verbo para fins de conjugação (Ex: **desencaminh**amos); já o segundo é a parte básica de uma palavra qualquer (Ex: desen**caminh**amos).



POS Tagging

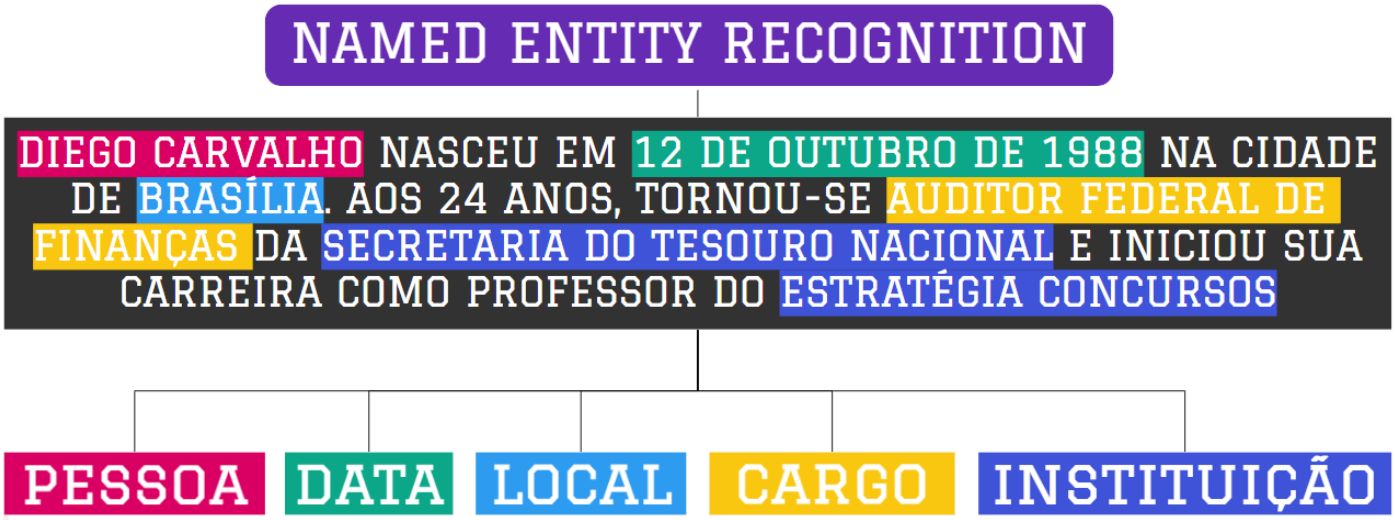
POS Tagging (Part-Of-Speech Tagging[3]) é o processo de identificar classes gramaticais em um texto com o intuito de complementar o processo de extração de palavras relevantes. Por exemplo: "*Cheguei ontem e apaguei*" pode ser representada como "*Cheguei/V ontem/ADV e/CJ apaguei/V*" (**V** = Verbo, **ADV** = Advérbio e **CJ** = Conjunção). Por exemplo: conforme vimos, em alguns casos é difícil para a máquina distinguir substantivos de verbos.



Named Entity Recognition

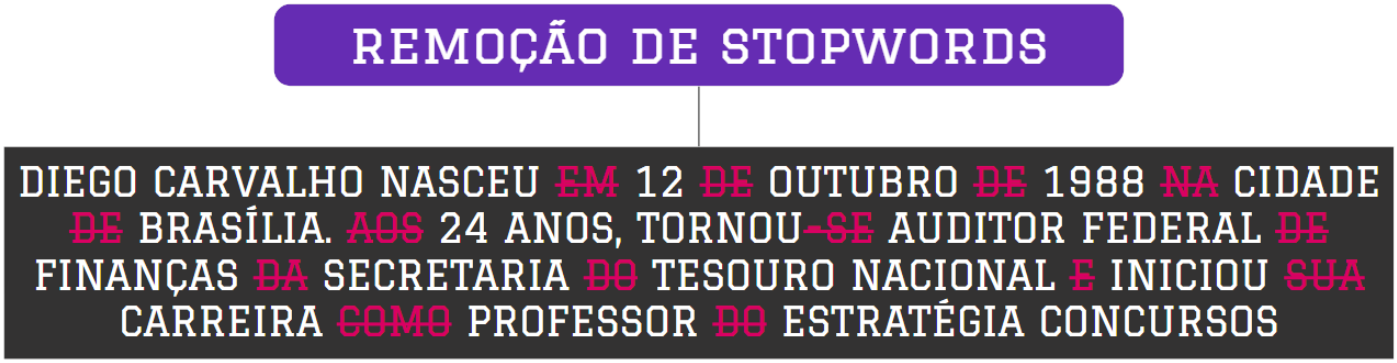
[3] Poderia ser traduzido como Marcação de Classes Gramaticais.

O Reconhecimento de Entidades Nomeadas é o processo de reconhecer entidades em um texto, tais como pessoas, datas, organizações, localizações, entre outros. *Como é, Diego?* Vejam no exemplo seguinte: algumas palavras foram reconhecidas como entidades, tais como nomes, datas, organizações, eventos, nacionalidades e localidades. *Vocês se lembram do Grafo de Conhecimento?* Isso aqui ajuda a criá-lo! :)



Stopwords Removal

Stopwords são palavras que podem ser consideradas de pouco valor para o entendimento do sentido de um texto, isto é, palavras semanticamente irrelevantes. Em geral, trata-se de artigos, preposições, pronomes e conjunções (Ex: as, e, que, os, de, para, com, sem, aquele, etc). Essas palavras podem ser ignoradas com segurança, realizando uma pesquisa em uma lista predefinida de palavras-chave, reduzindo o ruído e melhorando o desempenho.



Encerramos as técnicas de pré-processamento! Note que resolvemos vários problemas, mas ainda não sabemos como lidar com ambiguidade, contextos diferentes, etc. Vamos mais à frente...

Representação de Texto

RELEVÂNCIA EM PROVA: BAIXA

REPRESENTAÇÃO DE TEXTO

Trata-se do processo de transformar texto em estruturas ou representações numéricas que podem ser usadas por algoritmos de aprendizado de máquina para processar e analisar o texto. Isso inclui técnicas como tokenização, lematização, remoção de palavras irrelevantes, entre outros. A representação de texto é uma etapa importante em muitas tarefas de processamento de linguagem natural, como classificação de texto e análise de sentimento.

Texto é armazenado no computador como uma sequência de caracteres. Para que ele possa ser utilizado para aproximação semântica, é preciso representá-lo de forma a facilitar a identificação de padrões estatísticos. Dentre as formas de representação de texto mais comuns utilizadas atualmente, a principal é a vetorização. *O que é isso, Diego?* Trata-se de uma forma de representar palavras para que um computador consiga entendê-las

A vetorização faz isso atribuindo a cada palavra um valor numérico e, em seguida, representando as palavras por esses números. Isso permite que o computador compare palavras diferentes e reconheça quando duas palavras têm significados semelhantes, dentre outras dezenas de funcionalidades. Essa vetorização pode ocorrer sem contexto (BOW e TF-IDF) ou com contexto (N-Gramas). Vamos ver cada uma delas...

Representação Bag of Words

RELEVÂNCIA EM PROVA: BAIXÍSSIMA

BAG OF WORDS (SACOLA DE PALAVRAS)

Trata-se de um método de representação de texto como dados numéricos. Envolve atribuir um valor numérico a cada palavra em um documento de texto, com base em sua frequência de ocorrência no documento. Os valores numéricos são usados para criar uma matriz esparsa, que pode ser usada para analisar o documento em busca de padrões ou tendências.

Uma das principais abordagens para representar texto como um vetor é o BOW - Bag of Words (Saco de Palavras). *O que é isso, Diego?* Trata-se do processo de contar todas as palavras de um texto. Basicamente, cria-se uma matriz de ocorrência[4] para a frase ou documento, desconsiderando a gramática e a ordem das palavras. A frequência de ocorrência das palavras é utilizada como um recurso para o treinamento de um classificador e armazenada dentro de um vetor.

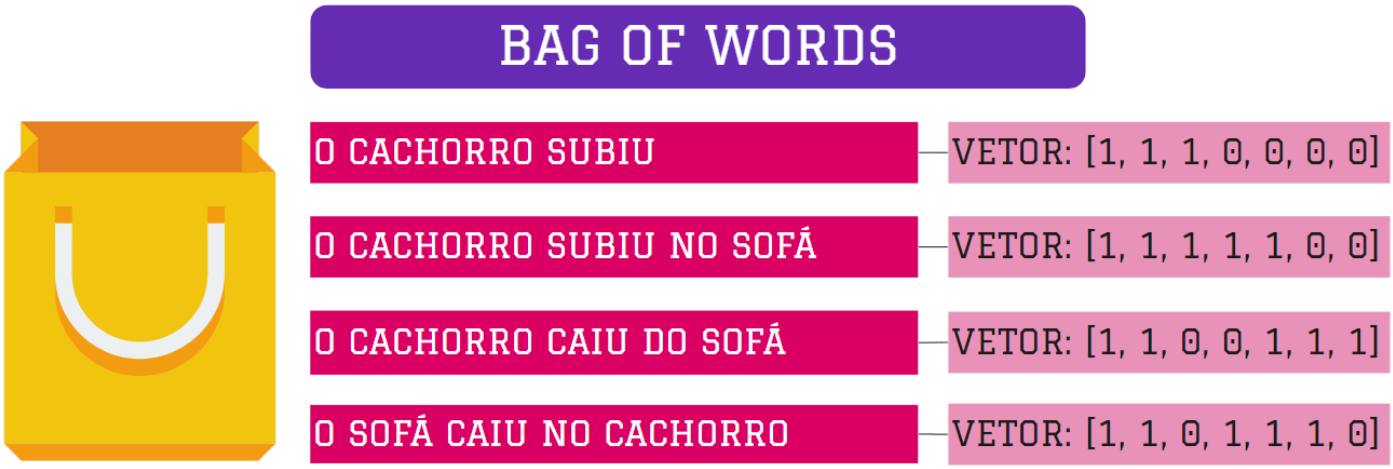
[4] Também chamada de Matriz de Co-ocorrência ou Matriz de Termo-Documento.

O nome desse processo é vetorização, isto é, um texto é representado através de um vetor numérico em que cada posição recebe a frequência referente a ocorrência da palavra no texto. Ex:

```
O cachorro subiu
O cachorro subiu no sofá
O cachorro caiu do sofá
O sofá caiu no cachorro
```

Primeiro, vamos definir nosso vocabulário! Ele é formado por todas as palavras encontradas no conjunto de documentos, quais sejam: "*o*", "*cachorro*", "*subiu*", "*no*", "*sofá*", "*caiu*", "*do*".

Documento/Frase	o	cachorro	subiu	no	sofá	caiu	do
O cachorro subiu	1	1	1	0	0	0	0
O cachorro subiu no sofá	1	1	1	1	1	0	0
O cachorro caiu do sofá	1	1	0	0	1	1	1
O sofá caiu no cachorro	1	1	0	1	1	1	0



Pronto! Agora nós vetorizamos cada documento com um vetor de comprimento fixo de sete posições conforme podemos ver a seguir:

```
O cachorro subiu:          [1, 1, 1, 0, 0, 0, 0]
O cachorro subiu no sofá:  [1, 1, 1, 1, 1, 0, 0]
O cachorro caiu do sofá:   [1, 1, 0, 0, 1, 1, 1]
O sofá caiu no cachorro:   [1, 1, 0, 1, 1, 1, 0]
```

Observe que perdemos informações contextuais, como a ordem das palavras. Por outro lado, esses vetores agora podem ajudar no treinamento de classificadores de texto. Não fará muito sentido agora, mas tudo fará mais sentido mais à frente. Aqui é importante salientar que essa técnica não é perfeita, dado que a frequência de ocorrência de palavras pode não significar muita coisa. *Já imaginou se eu aplico essa técnica em um documento muito grande?*

As palavras mais frequentes serão: *a*, *e*, *de*, *o*, *isso*, *para*, *que*, *não*, *na*, *no*, *até*, *com*, etc. Logo, partimos do princípio que – se uma palavra aparece com muita frequência – ela é menos relevante do que palavras que aparecem com baixa frequência. Para resolver esse tipo de problema, existem técnicas como TF-IDF (que utiliza um peso de acordo com o inverso da frequência); e PPMI (que utiliza probabilidade para o cálculo da frequência).

Na etapa de teste, o modelo treinado é avaliado utilizando-se uma porcentagem da base de dados rotulada, conhecida como conjunto de teste. Se a acurácia preditiva obtida durante a avaliação do modelo for satisfatória, então ele passa a ser utilizado para fazer a predição das classes de novas instâncias. Em outras palavras, se eu encontrar um conjunto de classes que consiga abarcar a maioria das instâncias da minha base, essas classes vão ser a base para as novas instâncias.

Representação TF-IDF

TF-IDF

TF-IDF (Term Frequency–Inverse Document Frequency) é uma estatística numérica usada para medir a relevância de uma palavra para um documento em um corpus. Trata-se de uma combinação de duas métricas: Frequência de Termo (TF) e Inverso da Frequência do Documento (IDF). O primeiro mede a frequência com que uma palavra aparece em um documento, enquanto o segundo mede a importância de uma palavra para um documento em uma coleção. O TF-IDF é usado para medir a relevância de uma palavra para um documento em uma coleção e é comumente usado na recuperação de informações e mineração de texto.

TF-IDF (*Term Frequency – Inverse Document Frequency*) é uma medida estatística que busca avaliar a importância de uma palavra em um documento em relação a uma coleção de documentos ou corpus. Digamos que você tenha coletado um milhão de documentos sobre uma ampla gama de assuntos e você deseja criar um mecanismo de busca para pesquisá-los. Para isso, você precisa determinar o quão relevante cada palavra é para cada documento.

Por exemplo: se eu procurar por "física nuclear", o motor de busca precisa saber o quão relevante são as palavras "nuclear" e "física" para cada documento. Dito isso, precisamos definir TF e IDF:

- **TF = *Term Frequency* = Frequência de Termo**

*Vocês concordam comigo que, se uma palavra aparecer muitas vezes em um documento, então é **provável** que essa palavra seja **relevante** para esse documento?*

- **IDF = *Inverse Document Frequency* = Frequência Inversa do Documento**

*Vocês concordam comigo que, se uma palavra aparecer em muitos documentos diferentes, então é **improvável** que essa palavra seja **relevante** para qualquer um deles especificamente?*

Professor, o que você quer dizer com relevante? Opa... isso é importante! Podemos chegar a uma definição mais ou menos subjetiva impulsionada pela nossa intuição: a relevância de uma palavra é proporcional à quantidade de informação que ela dá sobre seu contexto (uma frase, um documento ou um conjunto de dados). Dessa forma, as palavras mais relevantes são aquelas que nos ajudariam, como humanos, a entender melhor um documento inteiro sem ler tudo.

Por exemplo: nós já sabemos que palavras relevantes não têm necessariamente uma relação com sua frequência, dado que as stopwords são bastante frequentes (Ex: "a", "o", "de", "com", "por"). Todas essas palavras ocorrem com alta frequência em qualquer documento, logo são pouco relevantes. Por outro lado, algumas palavras ocorrem com alta frequência em um documento específico, mas ocorre com baixa frequência em outros documentos.

O que podemos concluir? TF-IDF é uma métrica estatística que leva em conta a frequência de uma palavra em um documento específico, mas busca penalizar palavras que ocorrem com alta frequência em diversos documentos. *Professor, dá um exemplo aí?* Claro! Imagine que desejamos avaliar a relevância da palavra "cachorro" em um conjunto de documentos. Para tal, nós precisamos analisar quão frequente ela é em um documento e quão rara ela é em outros.

Como já vimos anteriormente, se uma palavra é muito frequente em diversos documentos, provavelmente ela é pouco relevante; se uma palavra é pouco frequente em diversos documentos, mas é muito frequente em um documento específico, provavelmente ela é muito relevante. Para analisar essa métrica de relevância de uma palavra, temos uma fórmula matemática bem simples apresentada a seguir:

$$\mathbf{TFIDF} = \text{TF x IDF}$$

Sendo que:

$$\boldsymbol{TF(t,d)} = \frac{\text{QUANTIDADE DE TERMOS } \mathbf{t} \text{ NO DOCUMENTO } \mathbf{d}}{\text{QUANTIDADE DE TERMOS NO DOCUMENTO } \mathbf{d}}$$

$$\boldsymbol{IDF(N,n)} = log\ \frac{\text{QUANTIDADE (N) TOTAL DE DOCUMENTOS}}{\text{QUANTIDADE (n) DE DOCUMENTOS QUE CONTÊM O TERMO } \mathbf{t}}$$

Agora é fácil de fazer a conta! Vamos informar os valores: basta substituí-los na fórmula para obter o resultado desejado. Bem... nós queremos obter uma métrica de relevância para o termo "cachorro" e sabemos que ele aparece 5 vezes em um documento específico que contém 50 termos. Além disso, sabemos que foram analisados 100 documentos (N) e a palavra "cachorro" apareceu em 20 desses documentos (n). Dito isso, temos que:

$$\boldsymbol{TF(t,d)} = \frac{\text{QUANTIDADE DE TERMOS } \mathbf{t} \text{ NO DOCUMENTO } \mathbf{d}}{\text{QUANTIDADE DE TERMOS NO DOCUMENTO } \mathbf{d}} = \frac{5}{50} = 0{,}10$$

$$\boldsymbol{IDF(N,n)} = log\ \frac{\text{QUANTIDADE (N) TOTAL DE DOCUMENTOS}}{\text{QUANTIDADE (n) DE DOCUMENTOS QUE CONTÊM O TERMO } \mathbf{t}} = log\ \frac{100}{20} = log\ 5 = 0{,}69$$

Agora basta multiplicar:

$$\textbf{TFIDF} = \text{TF x IDF} = 0{,}10 \text{ x } 0{,}69 = 0{,}069$$

O que podemos concluir sobre esse valor? Quanto mais próximo de zero (0), maior é o peso, logo mais relevante é o termo; quanto mais longe de um (0), menor o peso, logo menos relevante é o termo. Em suma: se uma palavra ocorre várias vezes em um documento, devemos aumentar sua relevância, porque ela deve ser mais significativa do que outras palavras que aparecem menos vezes no documento (TF).

Ao mesmo tempo, se uma palavra ocorre muitas vezes em um documento, mas também em muitos outros documentos, talvez seja porque essa palavra é apenas uma palavra frequente (*stop words*); e, não, porque é relevante ou significativa (IDF). Essa métrica é frequentemente utilizada como fator de ponderação na recuperação de informações e na mineração de dados. É possível calcular essa métrica também para diversas palavras em uma submatriz de documentos x termos. Exemplo:

$$\begin{bmatrix} \text{TFIDF}_{(Termo\ 1,\ Documento\ 1)} & \text{TFIDF}_{(Termo\ 2,\ Documento\ 1)} \\ \text{TFIDF}_{(Termo\ 1,\ Documento\ 2)} & \text{TFIDF}_{(Termo\ 2,\ Documento\ 2)} \end{bmatrix}$$

Representação n-Gramas

RELEVÂNCIA EM PROVA: BAIXÍSSIMA

N-GRAMAS

Trata-se de conjuntos de palavras ou termos adjacentes em um determinado texto ou fala. Eles podem ser usados para analisar o contexto e o significado de um determinado texto e são normalmente usados para criar recursos para algoritmos de aprendizado de máquina, como classificação de texto e análise de sentimento.

Eu tenho uma pergunta para vocês: *prever o futuro é fácil?* Vocês já devem estar achando que eu enlouqueci, mas essa é uma pergunta válida! É claro que prever o futuro não é uma tarefa fácil, no entanto é possível afirmar com alguma convicção que algumas previsões são mais difíceis do que outras! Por exemplo: prever o que uma pessoa vai dizer – dependendo do contexto – não é tão difícil assim e eu vou provar isso para vocês! Imagine a seguinte frase:

```
Numa folha qualquer, eu desenho um sol _____________
```

Eu não sei a idade de vocês, mas todo mundo já ouviu a música chamada Aquarela de um cara chamado Toquinho (estou velho, não é possível que eu estou explicando isso para alguém). Ora, essa frase é exatamente igual ao primeiro verso da música e que termina com: **amarelo**. Logo, se uma pessoa estiver escrevendo essa frase em uma mensagem no Whatsapp, é mais provável que a próxima palavra seja *amarelo* do que *geladeira*. *Vocês concordam comigo?*

Vejam que eu não estou afirmando que a próxima palavra será *amarelo*. A próxima palavra pode ser *geladeira*, mas vocês hão de convir que é mais provável que a próxima palavra seja *amarelo* do que *geladeira*. Em geral, pode-se afirmar que algumas palavras são mais prováveis do que outras dependendo do contexto. Logo, introduzimos o conceito de probabilidade para avaliar qual será a próxima palavra. *Querem ver outros exemplos?*

```
Uma vez Flamengo, sempre _____________
```

```
Havia uma pedra no meio do __________
```

```
Que não seja imortal, posto que é chama; mas que seja infinito enquanto __________
```

```
Para meio entendedor, meia palavra __________
```

Professor, então quer dizer que isso somente é válido para músicas, ditados, poemas, entre outros? Não, é possível calcular isso para qualquer frase! Vejamos...

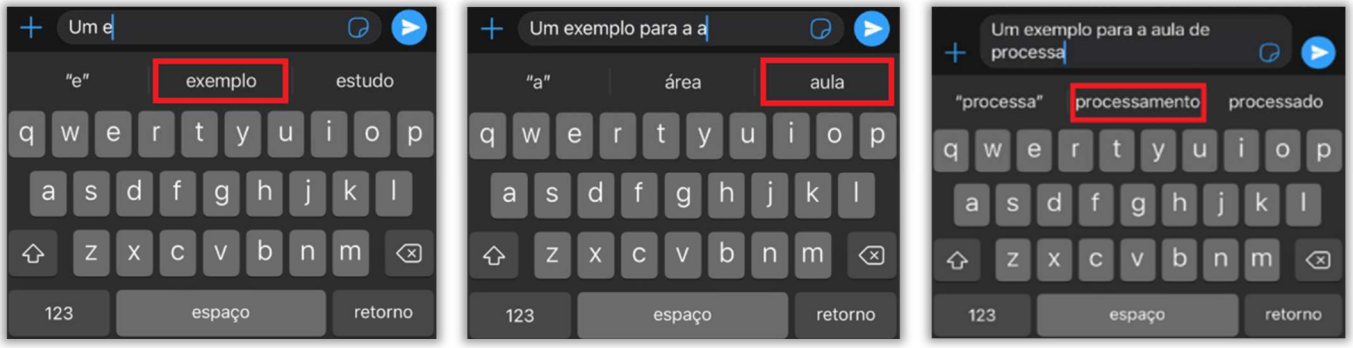
```
Quando terminar, você pode ligar para __________
```

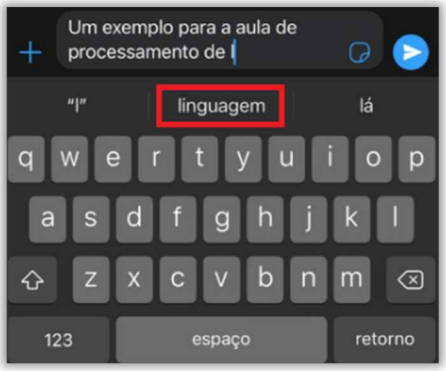
É mais provável que a próxima palavra seja *mim, ele, casa* do que *sapato, desde, gases.* Essa mesma ideia pode ser ampliada também para frases da seguinte forma:

```
As três raças são bastante suscetíveis a problemas respiratórios
```

```
Respiratórios raças a três suscetíveis são problemas bastante as
```

Qual das duas frases acima têm maior probabilidade de aparecer em um texto? Galera, é óbvio que a primeira frase tem uma probabilidade maior do que a segunda! *E por que desejaríamos prever palavras futuras ou atribuir probabilidades a frases?* Probabilidades são essenciais em qualquer tarefa em que tenhamos de identificar palavras em entradas de dados não estruturadas e ambíguas. O Teclado Inteligente de sistemas operacionais móveis já nos ajuda bastante:



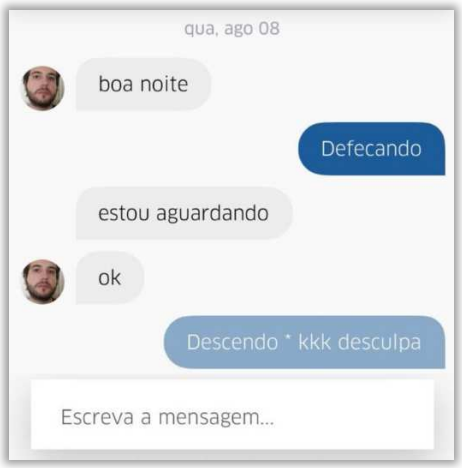


Uma outra aplicação bastante comum é em sistemas de transcrição de áudio. *Pessoal, quem aqui nunca errou a letra de uma música?* Vejam alguns exemplos:

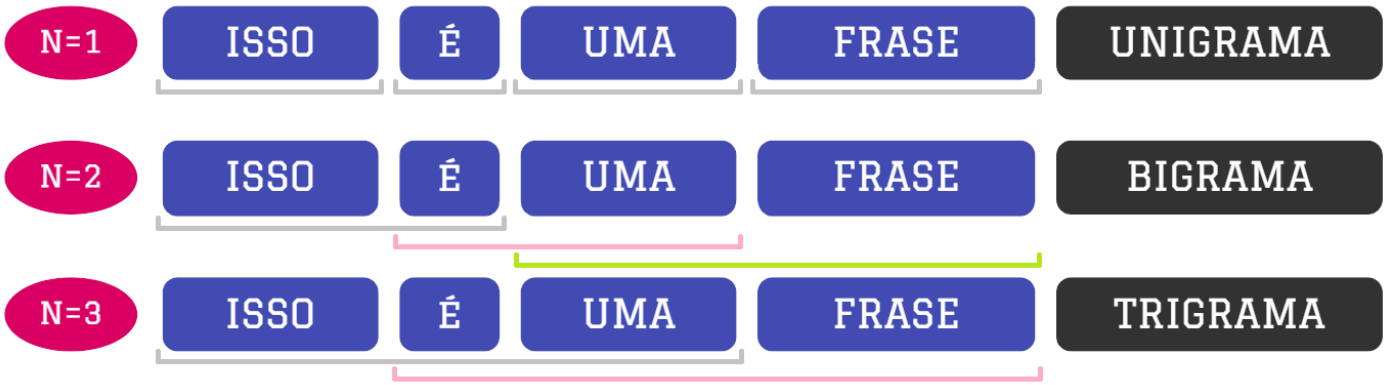
ERRADO	CORRETO
Na madrugada, vitrola rolando um blues, **trocando de biquini** sem parar...	Na madrugada, vitrola rolando um blues, **tocando B. B. King** sem parar...
Um abajur cor de **carne** e um lençol azul...	Um abajur cor de **carmim** e um lençol azul...
Eu perguntava **tudo em holandês** e te abraçava **tudo em holandês...**	Eu perguntava **Do You Wanna Dance** e te abraçava **Do You Wanna Dance...**
Entrei de **caiaque** no navio, entrei, entrei, entrei pelo cano...	Entrei de **gaiato** no navio, entrei, entrei, entrei pelo cano...
Quem sabe o príncipe virou um **sapo**, que vive dando no meu saco...	Quem sabe o príncipe virou um **chato**, que vive dando no meu saco....
E o teu futuro espelha essa grandeza, terra **dourada**...	E o teu futuro espelha essa grandeza, terra **adorada**...

Por vezes, nós – humanos – ouvimos algo, não temos certeza exatamente do que foi dito e cantamos errado! Os sistemas de transcrição de áudio também passam pelo mesmo problema, mas podem utilizar algoritmos para avaliar quais palavras são mais prováveis dentro de determinado contexto. Na frase a seguir, o algoritmo pode considerar mais provável que seja "*belos olhos*" do que "*cebola alhos*".

Outra aplicação interessante também é na correção gramatical a fim de auxiliar a corrigir erros em nosso idioma e também em sistemas de tradução de idiomas (Ex: Google Translator).



É claro que nem sempre dá certo e o corretor ortográfico acaba colocando a gente em situações indelicadas. Dito isso, existe um modelo de atribuição de probabilidades a sequências de frases ou palavras chamado n-grama (ou *n-gram*). Um n-grama é uma sequência de *n* palavras: um 2-grama (ou bigrama) é uma sequência de duas palavras; um 3-grama (ou trigrama) é uma sequência de três palavras; um 4-grama (ou tetragrama) é uma sequência de quatro palavras; e assim por diante[5].



Vamos relembrar um pouquinho de probabilidade do ensino médio? Prometo que é pouca coisa! Nós queremos calcular a probabilidade de uma certa palavra dada uma frase. Na formalidade matemática, podemos afirmar que desejamos calcular P(A|B), isto é, a probabilidade de uma palavra A dado um contexto B. *Como assim, Diego?* Vejamos um exemplo: vamos supor que o contexto seja "*Eu certamente vou me tornar um servidor*" e a palavra seja "*público*". Logo, temos que:

$$P(A|B) = P(\text{“público”} \mid \text{“Eu certamente vou me tornar um servidor”})$$

Uma maneira de estimar essa probabilidade é a partir de contagens de frequência relativa. Por exemplo: pegue um *corpus* muito grande, conte o número de vezes que vemos "*Eu certamente vou me tornar um servidor*", e conte o número de vezes que isso é seguido pela palavra "*público*". Isso

[5] Nessa aula, vamos nos focar apenas em palavras (e, não, em caracteres).

seria o equivalente a responder, de todas as vezes que vimos o contexto B, quantas vezes ele foi seguido pela palavra A.

Com um *corpus* bem grande (Ex: Web), nós podemos calcular essa frequência e estimar a probabilidade. Essa maneira de estimar probabilidade funciona em alguns casos, mas não é tão boa em nos dar prováveis estimativas em outros casos. Lembrem-se: línguas são muito criativas, novas frases são criadas o tempo todo e nós não vamos sempre ser capazes de contabilizar frases inteiras. Note que essa frase não retorna nenhum resultado no maior buscador do mundo:

Parece uma frase fácil de achar em um *corpus* tão grande quanto a web, mas lembrem-se que as aspas buscam frases inteiras exatas e, no exemplo acima, temos um 7-grama. Quanto maior o valor de *n*, mais difícil encontrar correspondências! Dito isso, os cientistas criaram uma maneira mais inteligente: *e se, ao invés de tentar calcular a probabilidade de uma determinada palavra dado um contexto grande, nós tentarmos calcular uma aproximação baseado apenas nas últimas palavras?*

Dito de outra forma, nós podemos utilizar um modelo de bi-gramas (2-grama) para calcular a probabilidade aproximada de uma palavra dada a palavra anterior[6]. Logo, teríamos que:

```
Bi-grama:  P(A|B) = P("público" | "servidor")
```

Professor, isso dá certo? Sim, por conta da Propriedade de Markov! O matemático Andrei Markov calculou que modelos probabilísticos podem prever a probabilidade de ocorrência de alguma palavra ou caractere futuro sem ter que olhar muito para trás. *Como é, Diego?* Ele calculou que a probabilidade de ocorrência de um 7-grama, por exemplo, é próxima de uma 2-grama. Logo, nós não precisamos olhar para todo contexto – basta olhar a palavra imediatamente anterior.

Eu não vou entrar aqui nos meandros matemáticos de cálculo de probabilidade de eventos independentes porque acredito que não faz sentido dentro do nosso contexto, dado que isso nunca foi cobrado em prova. De toda forma, saibam que – apesar de ser um modelo relativamente simples – existe toda uma sorte de cálculos matemáticos para chegar a uma boa estimativa de probabilidade. *E como saber se nosso qual modelo tem um bom desempenho?*

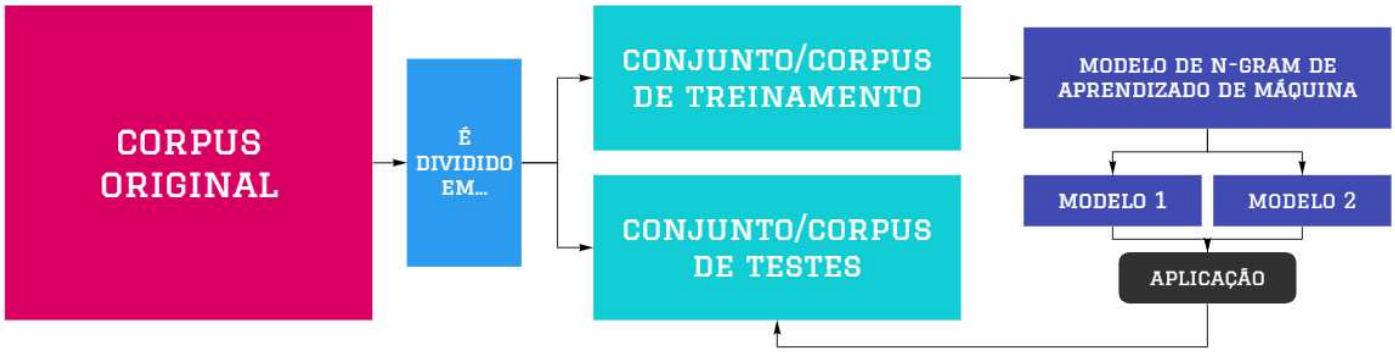
[6] Por vezes, é mais útil utilizar tri-gramas por conta do uso comum de artigos, preposições, pronomes, etc junto de substantivos.

A melhor maneira colocar dois ou mais modelos para rodar em uma aplicação real e depois analisar o resultado – também chamado de **avaliação extrínseca**. Por exemplo: podemos fazer um discurso em um sistema de transcrição de áudio e analisar se cada modelo transcreveu corretamente o que foi dito no discurso. *Como?* Muito simples: nós podemos avaliar quantas palavras cada modelo transcreveu de forma errada – o que errou menos é o melhor modelo. Só tem um problema...

Esse método consome muito dinheiro e tempo (pode durar dias ou semanas), dado que teríamos que rodar o modelo com diversos textos de diversos tamanhos com vozes de gêneros diferentes, sotaques diferentes, velocidade de fala diferentes, entre outros. *E qual é a alternativa, professor?* Podemos fazer uma avaliação intrínseca, que não envolve colocar o modelo para rodar na prática como na avaliação anterior – é independente de aplicação.

Como acontece com muitos dos modelos estatísticos, as probabilidades de um modelo de n-gram vêm do *corpus* em que é treinado – chamado de conjunto/corpus de treinamento. Podemos, então, medir a qualidade de um modelo de n-gram por seu desempenho em alguns dados chamados de conjunto/corpus de teste. Ao testar dois ou mais modelos, aquele que gerar um resultado mais próximo ao próprio conjunto de teste será considerado o melhor modelo. *E como funciona?*

Tudo começa com um *corpus* sendo dividido em *corpus* de treinamento e *corpus* de teste; depois nós utilizamos os dados do *corpus* de treinamento para treinar a máquina através de algoritmos de aprendizado de máquina; em seguida, comparamos o quão bem os dois modelos treinados se ajustam ao conjunto de teste. Vamos imaginar o cenário de transcrição de áudio que exemplificamos em páginas anteriores...



O *corpus* original é um conjunto de discursos em áudio. O primeiro passo é selecionar 90% desses discursos para serem utilizados no treinamento da máquina, isto é, vamos apresentar um discurso aos nossos modelos n-grama para que ele transcreva o resultado e vamos ajustando os parâmetros de modo que os modelos fiquem satisfatórios. Depois disso, nós colocamos o modelo para rodar nos outros 10% dos discursos em áudio. *E onde entra a probabilidade, professor?*

Na prática, a probabilidade não é utilizada na avaliação intrínseca, mas – sim – uma variante chamada perplexidade. *O que é isso?* A perplexidade é a probabilidade inversa de um conjunto de testes normalizada pelo número de palavras. Eu não vou detalhar esse ponto porque envolve uma

matemática mais aprofundada, então basta saber que a avaliação intrínseca utiliza a perplexidade para avaliar o melhor modelo.

Galera, a teoria sobre n-gramas é enorme, complexa e profunda! Poderíamos falar muito mais sobre esse assunto, mas acho que isso daqui já é o suficiente...

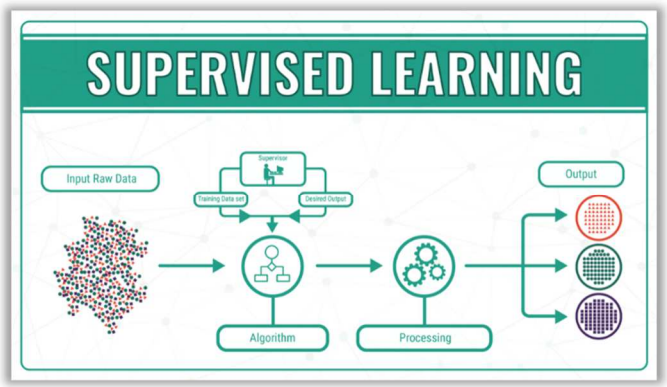
Classificação de Textos

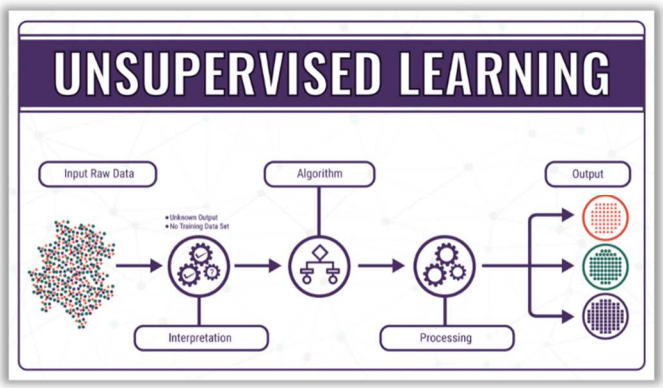
RELEVÂNCIA EM PROVA: MÉDIA

CLASSIFICAÇÃO DE TEXTOS

Trata-se do processo de atribuir um determinado documento de texto, como uma frase, parágrafo ou artigo curto, a uma ou mais categorias ou classes predefinidas. Isso é feito usando algoritmos de aprendizado de máquina supervisionados que são treinados em documentos de texto rotulados. O objetivo é determinar a categoria de um determinado documento de texto e usar essa informação para melhorar a precisão das tarefas de processamento de linguagem natural de um sistema. Aplicações comuns de classificação de texto incluem análise de sentimento, detecção de spam e categorização de documentos.

Antes de detalhar esse assunto, é importante falarmos sobre a diferença entre aprendizado supervisionado e aprendizado não-supervisionado. É bem simples: uma técnica de aprendizado supervisionado é aquela que necessita de supervisão ou interação com um ser humano, enquanto uma técnica de aprendizado não supervisionado não necessita desse tipo de supervisão ou interação. *Calma que ficará mais claro...*



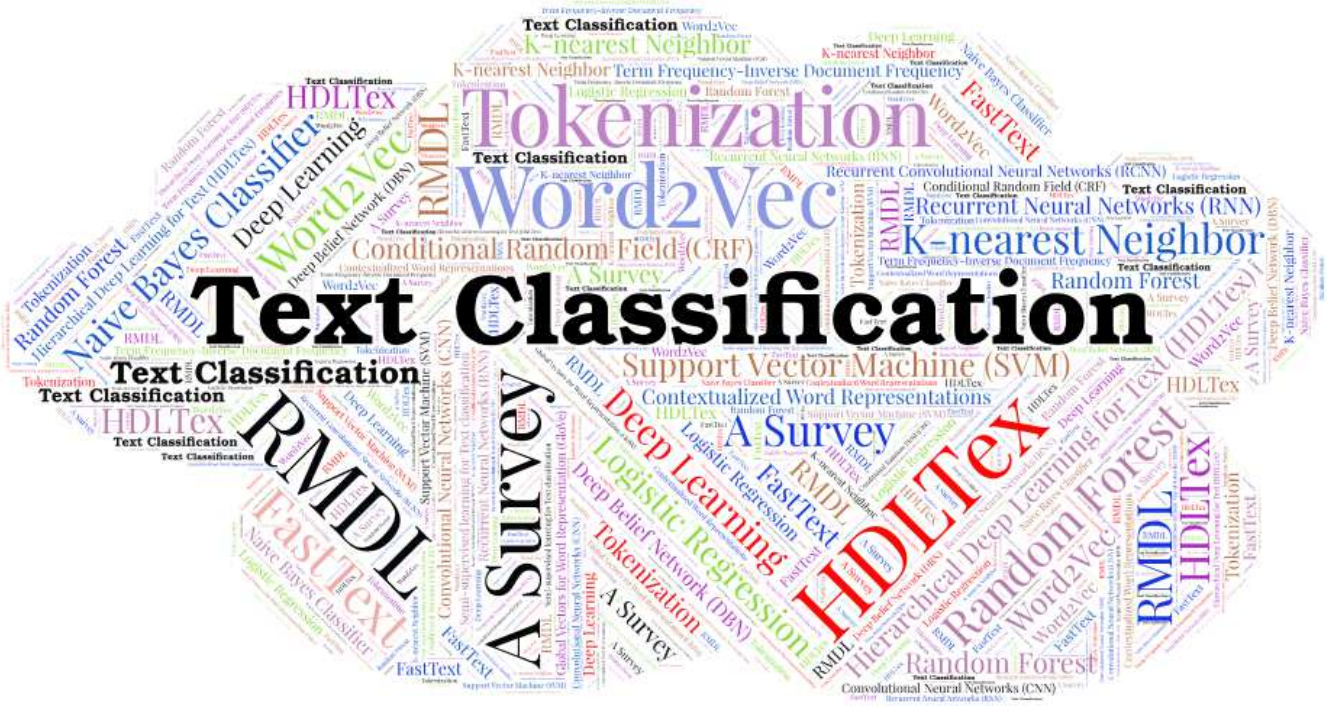


No aprendizado supervisionado, um ser humano alimenta o algoritmo com categorias de dados de saída de tal forma que o algoritmo aprenda como classificar os dados de entrada nas categorias de dados de saída pré-definidas. Vejam na imagem seguinte que há um conjunto de pontinhos, sendo que eu desejo classificá-los de acordo com as suas cores em vermelho, verde e roxo (o ser humano pode definir outra categoria como: pontinhos cujo nome da cor começa com a letra V ou R).

Note que – de antemão – eu já defini quais serão as categorias de saída, logo o algoritmo irá receber os dados brutos de entrada, irá processá-los e aprenderá a classificá-los em cada uma das categorias de saída que eu defini inicialmente. **Se um ser humano interferiu com o algoritmo pré-definindo nas categorias de saída do algoritmo, o algoritmo utilizou um aprendizado supervisionado porque ele aprendeu a categorização, mas com o auxílio de um ser humano.**

No aprendizado não supervisionado, um ser humano não alimenta o algoritmo com categorias de dados de saída pré-definidas. Vejam na imagem anterior que há um conjunto de pontinhos e o algoritmo – por si só – interpreta esses dados de entrada em busca de similaridades, padrões e características em comum, realiza o processamento e ele mesmo os categoriza sem nenhuma interferência humano durante o processo. **Logo, é chamado de aprendizado não supervisionado!**

Dito isso, agora vamos falar sobre a classificação de textos! Parece contraintuitivo, mas existem mais dados não-estruturados do que dados estruturados no mundo. Os dados não estruturados na forma de texto, como chats, e-mails, mídias sociais, etc podem ser uma fonte rica de informações, mas – devido à sua natureza – pode ser difícil extrair ideias dele. A classificação de texto é uma das tarefas de aprendizado supervisionado usada para ajudar a resolver esse problema.



Trata-se do processo de atribuição pré-definida (dado que se trata de um aprendizado de máquina supervisionado) de categorias a documentos baseado em seus conteúdos a fim de estruturar e analisar textos de maneira automática, rápida e econômica. É uma das tarefas fundamentais no Processamento de Linguagem Natural com amplas aplicações em análise de sentimento, detecção de spam, detecção de intenção e outros.

Uma vez constituída uma representação vetorial de um texto, seja por meio do BOW seja por meio do TF-IDF, a matriz numérica resultante pode ser utilizada como dado de entrada para qualquer algoritmo de aprendizado de máquina – inclusive de forma supervisionada com variável alvo (*target*) categórica. Em suma, uma vez que o texto está sendo representado como um vetor de números, é possível realizar classificações (ou regressões) como se fossem dados quaisquer.

Nesse sentido, as técnicas e conceitos vistos no contexto de aprendizado de máquina continuam válidos na classificação de textos no processamento de linguagem natural (Ex: validação e avaliação por separação de conjunto de dados de treino e de teste; *underfitting/overfitting*; regularização e otimização de hiperparâmetros; redução de dimensionalidade; modelos lineares, árvores de decisão, redes neurais, entre outros. Existem três abordagens de classificação:

- **Baseada em Regras**

Textos são separados em um grupo organizado utilizando um conjunto de regras. *Quem define essas regras?* Um ou mais usuários analisam uma amostra dos textos manualmente e criam regras para classificar dados. Por exemplo: eu desejo pegar um edital de concurso que não separou o conteúdo por disciplinas e devo classificar o conteúdo programático de acordo com cada matéria. Em primeiro lugar, eu devo definir quais serão minhas classes.

Vamos supor que eu tenha escolhido: português, direito administrativo, informática e raciocínio lógico. Abaixo seguem algumas regras que eu poderia criar para minha classificação:

```
Se uma frase contiver a palavra “verbo”, classificar como Português
...
Se uma frase contiver a palavra “nulidade”, classificar como Direito Administrativo
...
Se uma frase contiver a palavra “computador”, classificar como Informática
...
Se uma frase contiver a palavra “probabilidade”, classificar como Raciocínio Lógico
...
```

Eu posso criar várias regras dessas que nos ajudariam a classificar corretamente o conteúdo programático do edital em classes pré-definidas. Essa abordagem tem algumas desvantagens: exige um esforço manual que pode ser exaustivo no caso de documentos extremamente grandes e pode gerar falsos-positivos, principalmente por conta da polissemia (palavras com mais de um significado). *E a abordagem baseada em aprendizado de máquina?*

- **Baseada em Aprendizado de Máquina**

Ao utilizar o processamento de linguagem natural, a classificação de texto pode analisar automaticamente um texto e, em seguida, atribuir um conjunto de categorias predefinidas (também chamadas de rótulos, classes, *tags* ou *labels*) com base em seu contexto para o treinamento de um algoritmo de aprendizado de máquina que tem a função de aprender com os dados apresentados e predizer os rótulos de novas instâncias.

O processo de classificação é dividido em duas etapas: extração e teste. Na etapa de extração, utiliza-se uma amostra representativa de uma base de dados, conhecida como conjunto de treino, para extrair um modelo de classificação. Seu objetivo é encontrar um classificador que identifique a correlação entre as características (*features*) dos dados e suas respectivas classes. Para tal, é necessário estruturar o texto como um vetor numérico.

Em suma: algoritmos de aprendizado de máquina permitem inferir a classificação correta sem que usuários necessitem listar todas as regras manualmente. É claro que o algoritmo precisa de um treinamento inicial, mas – após o treinamento – ele é capaz de realizar a classificação correta mesmo de palavras novas que não estavam em seu treinamento. Esse algoritmo permite extrapolar essa classificação para outras palavras sem depender de novos treinamentos.

- **Abordagem Híbrida**

Por fim, existe também a abordagem híbrida que basicamente utiliza técnicas de ambas as abordagens – não muito o que acrescentar aqui!

E quais são as principais limitações da classificação de texto? Uma das principais limitações é que treinar um classificador de texto costuma ser o gargalo de treinamento. *Como assim, Diego?* A classificação de texto é uma atividade de aprendizado supervisionado, logo é necessário que alguém indique as categorias para o conjunto de dados de treinamento. Nem sempre há quantidades suficientes de exemplos de dados classificados para realizar o treinamento.

É verdade que modelos simples como regressão logística ou árvore de decisão treinados sobre textos vetorizados por BOW ou TF-IDF costumam dar bons resultados para tarefas simples mesmo com poucos dados. Por outro lado, para tarefas mais complexas (que requerem aproximar noções semânticas mais sutis, como ironia), podem-se utilizar modelos mais complexos de Deep Learning, treinados com quantidades muito maiores de dados. *Fechado?*

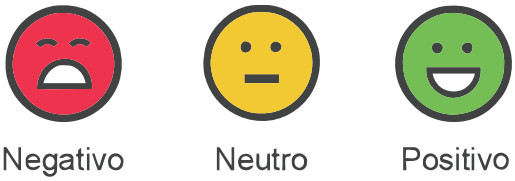
Análise de Sentimentos

RELEVÂNCIA EM PROVA: MÉDIA

ANÁLISE DE SENTIMENTOS

Também conhecida como Mineração de Opinião, trata-se do processo que busca identificar e extrair opiniões de um texto. Em geral, envolve a classificação de um trecho de texto como sentimento positivo, negativo ou neutro e pode ser usado para detectar opiniões sobre um determinado tópico, produto ou serviço em avaliações online, conversas em redes sociais e outras formas de dados de texto.

Opaaaaaaa... o papo agora é análise de sentimentos e eu adoro esse assunto! *Pessoal, vocês já pararam para pensar em quão avançada está a tecnologia atual?* A gente nem percebe, mas hoje todo mundo possui um pequeno dispositivo eletrônico que cabe no meu bolso. Esse aparelhinho pesa 150 gramas e permite fazer coisas sem ter sequer que manipulá-lo. Basta dizer algo como: "*E aí, Siri! Quem ganhou a última copa do mundo de futebol?*".



E esse pequeno conjunto de componentes eletrônicos nos responde de forma natural com uma voz feminina: "*O campeão da última copa do mundo de futebol foi a França em 2018*". Gente... isso é tão surreal que pareceria bruxaria algumas décadas atrás! O autor de ficção científica Arthur C. Clark (que escreveu o clássico 2001: Uma Odisseia no Espaço) inclusive formulou uma frase que ficou bastante conhecida que dizia assim:

```
Qualquer tecnologia suficientemente avançada é indistinguível de magia
```

Ele tem toda razão! Se mostrássemos essa tecnologia para alguém do início do século passado, eles certamente achariam que era bruxaria. No entanto, há perguntas que essa tecnologia ainda não responde. Por exemplo: "*E aí, Siri! Como eu estou me sentindo hoje? Como estão minhas emoções?*". Você vai me dizer que estou maluco; que esse tipo de pergunta ridícula não faz sentido nenhum; e que a máquina jamais vai conseguir me responder esse tipo de pergunta.

E se eu te disser que com o avanço de técnicas de inteligência artificial, aprendizado de máquina e análise de sentimentos, essa pergunta está cada vez mais próxima de ser respondida pela máquina?

Vamos entender isso melhor: pensem em uma nota de 0 a 10 para quanto uma pessoa gostou de um filme. Se ela diz que *amou* o filme, podemos considerar como uma nota 10; se ela diz que *gostou* do filme, podemos considerar como uma nota 7; e se ela diz que *odiou* o filme, podemos considerar como uma nota 0. A Análise de Sentimentos busca simplesmente utilizar técnicas de aprendizado de máquina para ensinar máquinas a extrair sentimentos da linguagem natural.

E como é o processo de análise de sentimentos? Uma análise básica de sentimentos, a partir de documentos de texto, segue um processo relativamente simples:

```
1. Divide cada documento de texto em partes menores como frases ou palavras;
2. Identifica cada sentimento associado a uma frase ou componente;
3. Atribui uma pontuação com a polaridade de sentimento a cada frase, de -1.0 até +1.0;
4. Em casos avançados, há combinação dessas pontuações com camadas de Deep Learning.
```

Todas as palavras que aprendemos, assim como suas utilizações em diferentes contextos, foram sendo construídas em nosso cérebro a partir de experiências. Isso possibilitou entendermos a força de cada adjetivo, recebendo sugestões e feedback de nossos professores, colegas e familiares ao longo do caminho. Logo, quando lemos uma frase, nosso cérebro baseia-se em conhecimento acumulado para identificar – em cada uma delas – um sentimento associado.

Dessa forma, conseguimos identificar e interpretar o grau de negatividade ou positividade no significado de uma sentença. É claro que isso geralmente acontece de forma subconsciente. Nós sabemos instintivamente que uma "*vitória esmagadora*" possui um sentimento positivo, enquanto que "*ser esmagado por um carro*" possui um sentimento negativo associado, por exemplo. Na análise de sentimentos de um computador, o funcionamento é quase o mesmo...

Professor, toda frase pode ser classificada como positiva ou negativa? Não, existe frases completamente neutras ou puramente informativas:

```
- Positivo  - Flamengo é o melhor time do mundo
- Negativo  - Meu dia está péssimo
```

```
- Neutro    - A Páscoa é na semana que vem
- Neutro    - Vire à esquerda em 400 metros
- Neutro    - Meu cachorro é um Golden Retriever
```

Então a taxa de assertividade desses algoritmos é altíssima? Ainda não! Por questões didáticas, eu dei exemplos muito tranquilos para vocês, mas existem problemas. Vejamos...

```
- Eu não gostei do jogo.
```

Note que há uma palavra com viés positivo (*gostei*), mas há outra negando esse viés (*não*). *E agora?* Nesses casos, há algumas técnicas que inserem um marcador de negação para ajudar o algoritmo.

```
- Eu não detestei o jogo.
```

Agora é pior ainda: temos uma dupla negação. *O que fazer?* Existe uma palavra negativa, mas existe uma negação da palavra negativa, logo o algoritmo tem que inverter o sentimento inicial...

```
- Detestar livros não é algo que eu aprecio.
```

Agora complicou mais ainda porque temos duas palavras negativas, uma palavra positiva e a ordem da frase também não nos ajuda.

```
- Obrigado, NET! Eu realmente amo esperar 30 minutos para ser atendido.
```

Agora ferrou de vez porque a pessoa está sendo irônica e, se tem uma coisa que computadores têm extrema dificuldade, é lidar com ironia (por essa razão, nem sempre a taxa de acerto é alta).

```
- Esse filme de terror tem um final perturbador.
```

No caso acima, foram utilizados termos negativos, porém com um sentido positivo – dado que finais perturbadores são desejados em filmes de terror.

```
- O hotel possui um ótimo quarto, mas o atendimento é horrível.
```

Note que se trata de um sentimento com polaridades diferentes sobre dois aspectos (quarto e atendimento), logo o computador tem dificuldade de categorizar.

```
- Aquele cara é zica!
```

Aqui a palavra zica pode ter um sentimento bom ou ruim dependendo do contexto, logo o algoritmo tem dificuldade de categorizar.

```
- Para bom entendedor, meia palavra basta.
```

Temos um ditado com uma palavra que possui sentimento positivo, no entanto esse ditado geralmente é utilizado em um contexto negativo.

```
- Iti Malia! Gaitei com o vídeo da promo cringe do BKing kkkk! #hitou #flopou
```

Nessa frase, temos de tudo: gírias, palavras incorretas, linguagem informal, abreviações e hashtags. *Estão vendo por que o algoritmo não consegue um altíssimo nível de assertividade?*

```
- Não quero dar spoiler, mas ando stalkeando e voltei a shippar o casal #BruMar 😍
```

O algoritmo tem dificuldade em lidar também com palavras de diferentes idiomas (estrangeirismos) em uma mesma frase. A vida do algoritmo não é nada fácil...

Existem diversas formas de classificar sentimentos, tais como: classificação por polaridade; classificação por escala; classificação por análise de elementos e aspectos; e classificação por análise de humor. Cada uma delas é indicada para um contexto específico (Ex: gestão de crise, panoramas gerais, estudos de satisfação de marcas, análise de opiniões predominantes para governos, entre outros). Não vamos entrar em detalhes sobre cada uma delas aqui.

Apesar de a análise de sentimentos ocorrer em sua maioria relacionado a frases, ela pode se dar em diferentes níveis de granularidade, de forma que – quanto menor a granularidade – mais específica é a classificação. Dessa forma, ela pode se dar em: nível de documento (análise de um texto como um todo); nível de sentença (análise de frases de um documento); nível de palavra ou dicionário (análise de palavras em frases); nível de aspecto (análise de aspectos dentro de uma frase).

Outro ponto importante é que existem sentenças subjetivas e objetivas. Uma sentença objetiva possui normalmente um fato ou uma informação (Ex: notícias de jornal), enquanto sentenças subjetivas expressam sentimentos pessoais e opiniões (Ex: postagens de usuários de redes sociais). Entender se um conjunto de dados possui mais sentenças objetivas ou subjetivas pode influenciar diretamente os resultados. Enfim...

A análise de sentimento nada mais é do que um caso particular de classificação de texto com apenas duas ou três classes, que refletem o sentimento de um texto. Interpretando atributos de uma árvore de decisão ou regressão logística treinada sobre um *dataset* de texto associado a sentimento, verifica-se que palavras tipicamente positivas (ótimo, maravilha...) são associadas à classe positiva e palavras tipicamente negativas (péssimo, horrível) são associadas à classe negativa.

Para evitar anotar manualmente um grande conjunto de textos em relação ao sentimento expresso, procura-se gerar essas anotações a partir de dados que já tenham alguma avaliação. Sabemos que, em geral, quando você faz uma boa crítica para um produto, você dá uma boa nota; e sabemos que, quando você faz uma crítica ruim para um produto, você dá uma nota ruim. Nesse contexto, temos o texto e sua respectiva avaliação (nota).

É claro que existem exceções (provavelmente provocada por erros no momento de dar a nota em um aplicativo), mas que geram muitas risadas. Há um compilado desse tipo de comentário que ocorre com frequência nas avaliações de comidas do iFood conforme podemos ver na imagem seguinte. É claro que esses são dados anômalos que confundem um pouquinho um algoritmo, mas que – como são excepcionais – não o inviabilizam.

Em síntese, podemos concluir que a análise de sentimento, também chamada de mineração de opinião, é um tipo de classificação de texto que analisa opiniões, sentimentos, avaliações, atitudes e emoções das pessoas em relação a entidades como produtos, serviços, organizações, indivíduos, eventos e seus aspectos, com o intuito de identificar sentimentos de forma automatizada a respeito de alguma entidade e traduzir as incertezas da emoção a fim de prover vantagem competitiva.

Atualmente, as aplicações de análise de sentimento são puramente comerciais. Nós vemos produtores de filmes utilizando análise de sentimento para avaliar o feedback do público sobre seus projetos recentes e vemos empresas incluindo esta tecnologia para avaliar como os consumidores estão reagindo aos seus produtos, mas futuramente, conforme essa tecnologia ficar melhor, poderemos ver aplicações a uma miríade de problemas.

Ela pode ser usada para fornecer ajuda para pessoas com problemas de saúde mental, dado que muitas delas encontram refúgio na internet e, portanto, com esta tecnologia será possível ajudar as pessoas que relutam em procurá-la. Além disso, ela pode ser usada para medir o radicalismo na internet, tornando-a mais segura para todos. Claro que também poderá ser usada para o mal, influenciando eleições como aconteceu nos EUA!

HEI, SIRI... JÁ É HORA DE SEGUIR PARA O PRÓXIMO ITEM?

Modelagem de Tópicos Latentes

RELEVÂNCIA EM PROVA: BAIXÍSSIMA

MODELAGEM DE TÓPICOS LATENTES

Trata-se de um tipo de aprendizado de máquina não supervisionado utilizado para analisar uma coleção de documentos e descobrir os tópicos subjacentes que estão presentes no texto. O modelo funciona encontrando padrões nas palavras usadas nos documentos e usa esses padrões para agrupar palavras semelhantes em tópicos. A modelagem de tópicos latentes pode ajudar a identificar tópicos em grandes quantidades de texto, como artigos de notícias, postagens de mídia social ou páginas da web, sem intervenção manual. Também pode ser usado para fornecer informações sobre o conteúdo semântico dos documentos.

O papo agora é sobre modelagem de tópicos latentes! De forma geral, podemos dizer que se trata da extração de tópicos[7] abstratos de uma coleção de documentos. *Para que isso?* Ora, uma das principais aplicações do processamento de linguagem natural é saber do que se tratam um grande número de documentos de texto diferentes. Só que é difícil para um ser humano ler todos esses documentos e extrair/compilar cada tópico.

Nesses casos, a modelagem de tópicos latentes é utilizada para extrair informações de documentos. Suponha que você esteja lendo alguns artigos em um jornal e, nesses artigos, as palavras "temperatura", "chuva" e "umidade" aparecem mais frequentemente do que quaisquer outras palavras. Por senso comum, você pode imaginar que esses artigos serão provavelmente sobre algo relacionado a clima.

A modelagem de tópicos latentes faz algo semelhante, porém de maneira estatística. É intuitivo imaginar que palavras de semânticas similares apareçam em contextos similares, ou seja, rodeadas pelo mesmo conjunto de palavras. Por exemplo: eu fiz uma macro no MS-Word para rodar nessa aula que eu estou escrevendo neste momento (e não acabei ainda) para me dizer quais são as palavras mais frequentes até agora. Vejam que interessante...

Por óbvio, as palavras com maior frequência foram as *stopwords*: "***de***" (1404), "***que***" (660), "***e***" (570), "***o***" (503) e "***um***" (478). Já considerando apenas as palavras relevantes, o ranking foi:

TERMO	FREQUÊNCIA
PALAVRA	350
TEXTO	191
DOCUMENTO	143
DADOS	130

[7] Tópico é um conjunto de palavras que ocorrem juntas mais frequentemente do que o esperado por acaso. Essas palavras são usadas para representar os "temas" subjacentes dos dados e obedecem a uma distribuição de probabilidade sobre o vocabulário.

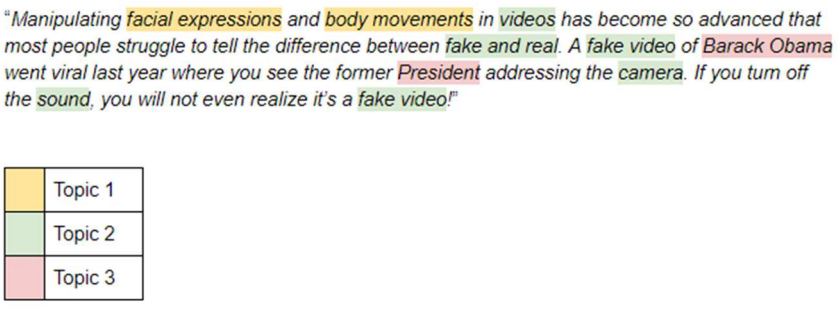
MODELO	99
LINGUAGEM	78
NATURAL	75
ANÁLISE	72
CONJUNTO	65
PROCESSO	53

Você há de concordar comigo que esse conjunto de palavras ocorre juntas com uma frequência muito maior quando tratamos do tema de processamento de linguagem natural do que quando falar de, por exemplo, direito constitucional. Da mesma forma que palavras ligadas a textos cujo tema é futebol terão uma frequência de ocorrência juntas muito maior do que em um texto cujo tema é, por exemplo, astronomia.

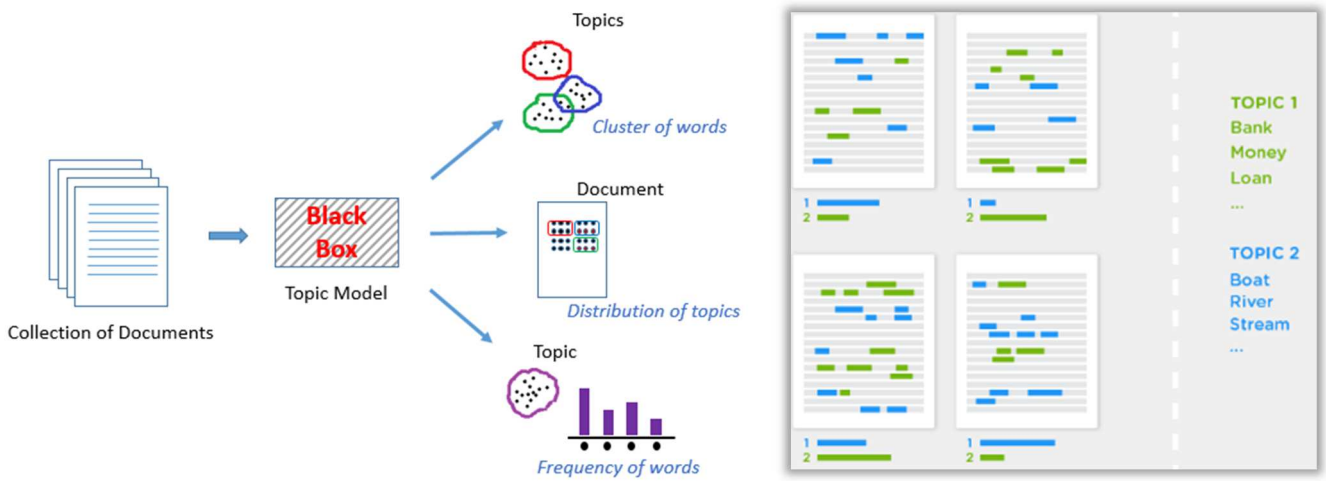
Essa é a base para a modelagem de tópicos latentes, mas evidentemente com maior complexidade. Em regra, temos um texto como um BoW transformado em uma matriz de frequências TF-IDF, que nos devolve as palavras que ocorrem frequentemente juntas (além de sua distribuição de probabilidade). Já eu fiz algo bem mais simples, apenas contabilizando as palavras mais frequentes do texto em um único documento.

Antes de seguir, é importante não confundir dois conceitos: "*modelagem de tópicos*" e "*classificação de tópicos*". Embora pareçam semelhantes, são processos totalmente diferentes: a classificação de tópicos é uma técnica de aprendizado de máquina supervisionada; já a modelagem de tópicos é não supervisionada. A classificação de tópicos frequentemente envolve classes mutuamente exclusivas, o que significa que cada documento é rotulado com uma única classe específica.

Por outro lado, a modelagem de tópicos não é mutuamente exclusiva, o que significa que um mesmo documento pode envolver diversos tópicos diferentes. Como a modelagem de tópicos funciona com base na distribuição de probabilidade, o mesmo documento pode ter uma distribuição de probabilidade espalhada por muitos tópicos. Vejam que na imagem apresentada a seguir, temos três tópicos distintos.



Um modelo de tópico identificará palavras semelhantes e as colocará em um grupo/tópico. O tópico mais dominante no exemplo acima é o Tópico 2, que indica que este texto é sobre vídeos falsos.



Note pela imagem à esquerda que o modelo de tópicos recebe uma coleção de documentos e retorna um agrupamento de palavras, a distribuição dos tópicos e a frequência de palavras. Esses tópicos têm uma certa distribuição em um documento e cada tópico é definido pela proporção de palavras diferentes que ele contém. Na imagem à direita, temos quatro documentos e vemos que o modelo agrupou as palavras em dois tópicos: Tópico 1 e Tópico 2.

Por outro lado, vejam que um mesmo documento contém palavras relacionadas a tópicos diferentes, justamente porque não é uma modelagem mutuamente exclusiva. Em síntese, podemos afirmar que a modelagem de tópicos latentes é um conjunto de técnicas não supervisionadas que permitem extrair tópicos abstratos a partir das palavras dos textos e agrupar os textos de acordo com esses temas.

No meu documento, eu fiz apenas a contabilização da frequência das palavras mais comuns, mas não montei nenhuma Matriz TF-IDF. De todo modo, a ideia aqui é analisar a estrutura dessa matriz a fim de descobrir quais palavras ocorrem juntas frequentemente de forma que a co-ocorrência dessas palavras formem um tópico significativo, sistemático e coerente. Bem, existem basicamente duas técnicas principais para implementação de modelagem de tópicos: LSA e LDA.

A LSA (*Latent Semantic Analysis*) é uma técnica utilizada para analisar as relações entre as palavras e frases dentro de um texto. Baseia-se na suposição de que as palavras usadas no mesmo contexto terão significados semelhantes e podem, portanto, ser usadas para identificar a semântica subjacente de um texto. Ela usa uma técnica matemática chamada decomposição de valor singular para identificar padrões comuns em diferentes documentos e textos.

Ao analisar os padrões, ela pode identificar tópicos e temas em um texto, permitindo uma melhor indexação e recuperação de documentos. Essa técnica é rápida e fácil de implementar e oferece resultados razoáveis, mas tem diversas desvantagens (Ex: não funciona bem em conjuntos de dados com dependências não lineares, é computacionalmente pesada, entre outros). O que é mais importante atualmente (inclusive em concursos) é a outra técnica...

O LDA (*Latent Dirichlet Allocation*) é um modelo estatístico generativo que permite que conjuntos de observações sejam explicadas por grupos não observados que explicam por que algumas partes dos dados são semelhantes. Por exemplo, se as observações são palavras coletadas em documentos, isso pressupõe que cada documento é uma mistura de um pequeno número de tópicos e que a presença de cada palavra pode ser atribuída a um dos tópicos do documento.

A ideia básica aqui é que os documentos são considerados misturas aleatórias de vários tópicos e os tópicos são considerados uma mistura de palavras diferentes. Agora suponha que você precise de alguns artigos relacionados a animais, você tenha milhares de artigos na sua frente e você realmente não saiba do que tratam esses artigos – sendo que ler todos esses artigos é realmente complicado para descobrir os artigos relacionados a animais. Vamos ver um exemplo disso:

Nós batemos o olho nas palavras de cada artigo e já sabemos do que provavelmente se trata cada um deles, mas se fossem milhares de artigos com dezenas de linhas, já ficaria bem mais difícil. O computador pode fazer isso com a ajuda do Latent Dirichlet Allocation (LDA). Ele basicamente roda o algoritmo (que envolve distribuição de probabilidade, portanto não vamos detalhar aqui) e chega à seguinte conclusão:

```
Artigo #1: 100% sobre Tópico X
Artigo #2: 100% sobre Tópico Y
Artigo #3: 100% sobre Tópico Z
Artigo #4: 75%  sobre Tópico X e 25% sobre Tópico Y
```

Note que ele não vai te dizer o que é X, Y e Z! Quem fará isso é o usuário, mas o algoritmo já facilitou muito a nossa vida. Agora nós podemos afirmar que X é Animal, Y é Genética e Z é Computador:

É claro que existe o fenômeno da polissemia, isto é, palavras que possuem mais de um significado e que podem confundir o algoritmo, mas isso pode eventualmente ser tratado. *E quais são as diferenças fundamentais entre ambas as técnicas?* LSA não assume qualquer pressuposto quanto à

distribuição probabilística de tópicos em documentos, logo é um modelo pouco interpretável; LDA assume que a distribuição de tópicos segue a Distribuição Probabilística de Dirichlet.

Agora para finalizar, uma pergunta que ninguém me fez até agora: por que diabos esse assunto se chama modelagem de tópicos... *latentes?* Ora, porque ele consegue identificar de maneira não supervisionada (ou seja, sem o auxílio de um supervisor, professor, orientador) tópicos dentro de uma imensidão de palavras diferentes. Tópicos que, por vezes, pareciam estar... escondidos, invisíveis, ocultos!

Aliás, eu vou me atrever a dar uma pequena explicação gramatical: não confundam "***latente***" com "***patente***". O primeiro termo se refere àquilo que existe de maneira subentendida, que não se manifestou, não evidente, escondida, oculta, adormecida, etc; já o segundo termo se refere àquilo que está claro, que não deixa dúvidas, evidente, cristalino, explícito, etc. *Bacana?* Estou fazendo esse apêndice apenas porque eu vejo muitos alunos errando com frequência :)

Semântica Vetorial

RELEVÂNCIA EM PROVA: BAIXÍSSIMA

SEMÂNTICA VETORIAL

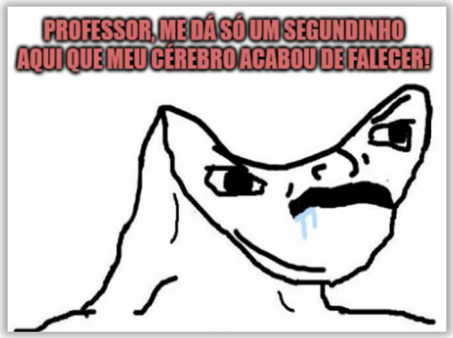
Trata-se de um tipo de análise semântica computacional que usa representações baseadas em vetores para palavras. Este método analisa o contexto das palavras em uma frase ou documento para criar um vetor numérico de valores com base na co-ocorrência de palavras nesse contexto. Esse vetor numérico, ou espaço vetorial, é então usado para representar o significado das palavras no texto. A semântica vetorial pode ser usada para executar tarefas como similaridade e classificação de palavras, modelagem de tópicos e análise de sentimentos.

Existe um ditado chinês que diz: "*Redes são para peixe; uma vez que você pegue o peixe, você pode esquecer a rede*". Nós podemos adaptá-lo para o nosso contexto da seguinte forma: "*Palavras são para significado; uma vez que você entenda o significado, você pode esquecer as palavras*". Pessoal, as palavras existem apenas, e tão somente, para representar partes do pensamento humano e por isso constitui uma unidade da linguagem humana – um significado.

No entanto, o significado de uma palavra não é único – ele dependerá do contexto em que a palavra é utilizada. Se eu entrar em um planetário espacial e pedir para me explicarem o que é um sol, eu terei uma resposta; se eu entrar em uma escola de música e fizer exatamente a mesma pergunta, eu terei outra resposta completamente diferente. Aliás, existe um postulado que fundamenta todo o ramo de semântica estatística chamado Hipótese Distributiva, que afirma que:

HIPÓTESE DISTRIBUTIVA

QUANTO MAIS SEMANTICAMENTE SEMELHANTES FOREM DUAS PALAVRAS, MAIS SEMELHANTES EM TERMOS DE DISTRIBUIÇÃO ELAS SERÃO E, PORTANTO, MAIS TENDERÃO A OCORRER EM CONTEXTOS LINGUÍSTICOS SEMELHANTES



Calma, pessoal... eu sei que parece difícil, mas eu juro que não é! Vejam só: em meados da década de 1950, pesquisadores verificaram que palavras sinônimas (Ex: *oculista* e *oftalmologista*) tinham a tendência de ocorrer em um mesmo contexto ou ambiente – geralmente próximas das palavras *olhos* ou *exames*. O que a hipótese distributiva diz é apenas que, se essas palavras tendiam a aparecer em contextos semelhantes, logo elas tendiam a ter significados semelhantes.

PALAVRAS QUE APARECEM EM CONTEXTOS SEMELHANTES TENDEM A TER SIGNIFICADOS SEMELHANTES

No pior cenário, caso duas palavras tenham contextos idênticos, então elas podem ser consideradas sinônimos. *Por que, Diego?* Por que a diferença de significado entre duas palavras é relativamente proporcional à diferença entre seus contextos. Isso parece complicado na teoria, mas na prática é muito fácil! Vejamos: *alguém aí sabe o que é um tesgüino?* Eu aposto que não! Agora leiam as quatro frases apresentadas a seguir:

- **A garrafa de tesgüino está sobre a mesa.**
- **Todo mundo gosta de tesgüino.**
- **O tesgüino pode te deixar bêbado.**
- **Tesgüino é feito de milho.**

E agora, deu uma clareada? Por meio desse contexto, já é possível concluir que tesgüino é um tipo bebida alcoólica feita de milho. *Como você concluiu isso?* Pesquisadores afirmam que o cérebro humano faz uma busca por outras palavras e analisa se elas se encaixariam nesses contextos. Se sim, elas são consideradas; se não, elas são eliminadas. Se nós pudéssemos representar isso, seria mais ou menos assim:

- **A garrafa de _____ está sobre a mesa.**
- **Todo mundo gosta de _____.**
- **O _____ pode te deixar bêbado.**
- **_____ é feito de milho.**

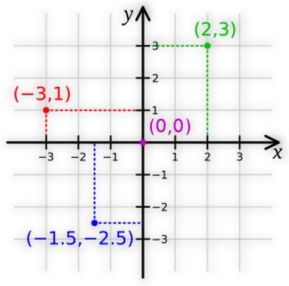
O cérebro humano faz uma busca rápida no vocabulário que conhecemos e verifica: (1) a palavra *alicate* se encaixa em algum desses contextos? Não, então é descartada; (2) a palavra *pamonha* se encaixa ema algum desses contextos? Sim, mas apenas no último – então é descartada por ser improvável; (3) a palavra vinho se encaixa em algum desses contextos? Sim, exceto no último – então é considerada por ser provável.

Claro que aqui é uma simplificação: seu cérebro faz isso de maneira absurdamente rápida e eficiente. O ponto aqui é: se eu encontrei duas palavras que podem ser utilizadas em diversos contextos semelhantes (Ex: vinho e tesgüino), então eu posso concluir – pela hipótese distributiva – que essas palavras têm significados semelhantes. *Quer ver uma palavra que se encaixaria em todos os contextos acima?* Cerveja! E adivinha só: tesgüino é uma cerveja criada pelo povo asteca no século XIV.

Agora vejam só: nós temos palavras, palavras têm significados e nós queremos representá-los de alguma maneira. Uma sugestão seria por meio de um vetor! *O que é um vetor, professor?* Trata-se

de uma estrutura de dados que armazena uma coleção de elementos que podem ser identificados por meio de um índice ou uma chave. A técnica que permite que computadores compreendam o significado de palavras por meio de vetores é conhecido como semântica de vetores.

Baseado nessa técnica, um computador pode capturar o significado de uma palavra por meio de sua distribuição no uso da linguagem, isto é, por meios das palavras ao redor (chamado de contexto ou ambiente gramatical). Um detalhe: o ato de embutir palavras dentro do espaço vetorial é conhecido como *Word Embedding*. E mais: vetores podem ter várias dimensões, como vetores unidimensionais, bidimensionais (matrizes), tridimensionais...



Uma propriedade interessante de vetores é que eles podem representar coordenadas. Ex: os pontos apresentados ao lado estão em um plano cartesiano de forma que possam ser representados por um vetor que armazena suas coordenadas no espaço. Note que rosa se encontra em [0, 0]; verde se encontra em [2, 3]; vermelho se encontra em [-3, 1]; e azul se encontra em [-1,5, -2,5]. Dito isso, é possível representar quão frequentemente palavras ocorre por meio de uma Matriz de Co-ocorrência.

Existem dois tipos dessa matriz: Termo-Documento e Termo-Termo. No primeiro caso, cada linha da matriz representa um termo e cada coluna representa um documento. Nesse contexto, uma matriz é um vetor bidimensional semelhante a uma tabela; um termo é uma palavra do vocabulário; e um documento é um registro escrito sobre qualquer assunto. Vamos ver um exemplo para ficar mais claro para vocês! Primeiro, vamos desenhar uma matriz...

TERMO/DOCUMENTO				

Nós sabemos que as colunas representam documentos. Dessa forma, os documentos da nossa matriz serão quatro peças de William Shakespeare: duas comédias e dois dramas[8]. Vejamos:

TERMO/DOCUMENTO	COMO GOSTAIS	NOITE DE REIS	JÚLIO CÉSAR	HENRIQUE V

[8] Em Processamento de Linguagem Natural, textos escritos e registros orais em linguagem natural que servem de base para análise é chamado de ***corpus*** e um conjunto de corpus é chamado de ***corpora***. Na prática, corpus são os documentos (também chamados de *datasets*).

Por fim, nós sabemos que as linhas representam termos do vocabulário. Em nosso contexto, escolhemos as palavras *batalha*, *bom*, *bobo* e *sagaz*:

TERMO/DOCUMENTO	COMO GOSTAIS	NOITE DE REIS	JÚLIO CÉSAR	HENRIQUE V
BATALHA				
BOM				
BOBO				
SAGAZ				

Ora, se a ideia é que essa matriz apresente a quantidade de ocorrências, então o conteúdo das células será a quantidade de vezes em que cada palavra aparece em cada peça:

TERMO/DOCUMENTO	COMO GOSTAIS	NOITE DE REIS	JÚLIO CÉSAR	HENRIQUE V
BATALHA	1	0	7	13
BOM	114	80	62	89
BOBO	36	58	1	4
SAGAZ	20	15	2	3

Finalizamos a Matriz Termo-Documento! Agora podemos verificar, por exemplo, que a palavra *bobo* aparece 58 vezes na peça *Noite de Reis* e que a palavra *sagaz* aparece 3 vezes *em Henrique V*.

Legal! Agora olha que interessante: uma matriz é formada por um conjunto de vetores. Nós podemos dizer, por exemplo, que cada coluna dessa matriz é um vetor:

TERMO/DOCUMENTO	COMO GOSTAIS	NOITE DE REIS	JÚLIO CÉSAR	HENRIQUE V
BATALHA	1	0	7	13
BOM	114	80	62	89
BOBO	36	58	1	4
SAGAZ	20	15	2	3
	VETOR 1	VETOR 2	VETOR 3	VETOR 4

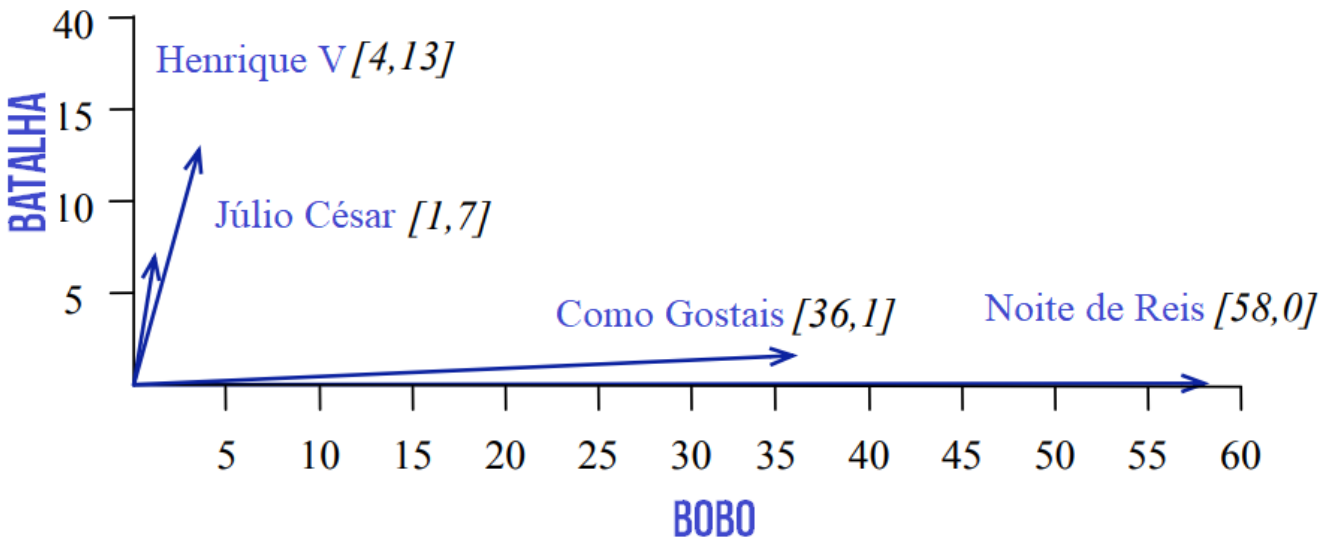
Vejam que maneiro: como uma matriz é formada por um conjunto de vetores, cada peça pode ser representada por um vetor de quatro posições. Vejamos:

```
- Como Gostais    = [1, 114, 36, 20]
- Noite de Reis   = [0, 80, 58, 15]
- Júlio César     = [7, 62, 1, 2]
- Henrique V      = [13, 89, 4, 3]
```

Mais maneiro ainda: eu posso pensar em um vetor para um documento como um ponto em um Espaço V-Dimensional[9]. *Como é, Diego?* No esquema acima, nós temos quatro pontos/palavras para cada documento, logo temos um espaço 4-Dimensional (ou tetradimensional) – sendo cada documento é uma dimensão. Como é complicadíssimo representar graficamente um espaço de quatro dimensões, vamos fazer uma pequena adaptação na nossa matriz termo-documento:

TERMO/DOCUMENTO	COMO GOSTAIS	NOITE DE REIS	JÚLIO CÉSAR	HENRIQUE V
BATALHA	1	0	7	13
BOBO	36	58	1	4
	VETOR 1	VETOR 2	VETOR 3	VETOR 4

Pronto! Agora temos apenas duas dimensões, logo podemos representar essa matriz como um plano cartesiano de Bobo x Batalha:

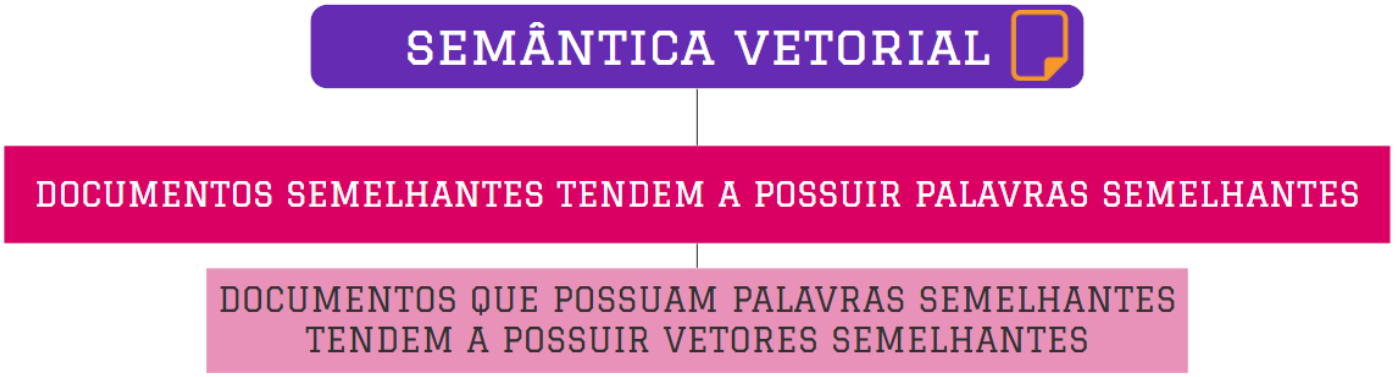


O que essa imagem está nos mostrando? Note que o termo *bobo* aparece mais vezes nas peças de comédia, enquanto o termo *batalha* aparece mais vezes nas peças de drama. Ora, isso faz todo sentido! Nós sabemos que dois documentos similares tendem a possuir palavras similares e se dois documentos tiverem palavras semelhantes, seus vetores de coluna tenderão a ser semelhantes. Vamos tentar verificar isso...

Os vetores para as comédias *Como Gostais* [1.114,36,20] e *Noite de Reis* [0,80,58,15] se parecem muito mais (mais *bobo* e *sagaz* do que *batalha*) do que *Júlio César* [7,62,1,2] ou *Henrique V* [13,89,4,3]. Isso fica claro com os números brutos: na primeira dimensão (*batalha*), as comédias têm números baixos e os dramas têm números altos. Note abaixo que os vetores em verde possuem números mais próximos e estão mais próximos no espaço vetorial; ao contrário dos vermelhos.

[9] Em um espaço V-Dimensional, V representa a quantidade de palavras no vocabulário.

```
- Comédia:        Como Gostais      = [1,  114, 36, 20]
- Comédia:        Noite de Reis     = [0,  80,  58, 15]
- Drama:          Júlio César       = [7,  62,  1,  2 ]
- Drama:          Henrique V        = [13, 89,  4,  3 ]
```



Galera, é claro que aqui é tudo uma simplificação – uma matriz termo-documento não teria apenas quatro linhas e colunas. De maneira mais geral, ela possuirá N linhas (uma para cada tipo de palavra no vocabulário) e X colunas (uma para cada documento). Lembrando que o tamanho do vocabulário é geralmente na casa das dezenas de milhares e o número de documentos também pode ser gigantesco (imaginem todas as páginas web, por exemplo).

Bem, nós criamos vetores de documentos (colunas), mas também é possível criar vetores de linhas (palavras) para representar seus significados. Vejamos:

TERMO/DOCUMENTO	COMO GOSTAIS	NOITE DE REIS	JÚLIO CÉSAR	HENRIQUE V	
BATALHA	1	0	7	13	VETOR 1
BOM	114	80	62	89	VETOR 2
BOBO	36	58	1	4	VETOR 3
SAGAZ	20	15	2	3	VETOR 4

Note que permanece sendo um vetor de quatro dimensões: *Batalha* = [1, 0, 7, 13], *Bom* = [114, 80, 62, 89], *Bobo* = [36, 58, 1, 4]; e *Sagaz* [20, 15, 2, 3]. Para documentos, vimos que documentos semelhantes têm vetores semelhantes, porque documentos semelhantes tendem a ter palavras semelhantes. Este mesmo princípio se aplica às palavras: palavras semelhantes têm vetores semelhantes porque tendem a ocorrer em documentos semelhantes.

A Matriz Termo-Documento, portanto, nos permite representar o significado de uma palavra pela quantidade de documentos que ela tende a ocorrer. De toda forma, em ambos os casos temos um eixo representando documentos e outro eixo representando palavras. Já a Matriz Termo-Termo (também chamada de Matriz Palavra-Palavra ou Palavra-Contexto) possui ambos os eixos representando palavras. *Como é isso, Diego?* Seria mais ou menos assim:

TERMO/TERMO	TERMO 1	TERMO 2	TERMO 3	TERMO 4
TERMO 1				
TERMO 2				
TERMO 3				
TERMO 4				

Essa matriz terá dimensionalidade |V| x |V| e cada célula armazenará o número de vezes que a palavra da linha e a palavra da coluna co-ocorrem no contexto de algum *corpus*. Eu vou fazer uma pequena pausa aqui: |V| representa a quantidade de palavras distintas em um conjunto de documentos. Ora, se meus documentos forem livros, imaginem a quantidade de palavras distintas existirão. Logo, imaginem quão absurdamente gigantesca será essa matriz...

Por que eu estou falando isso? Para enfatizar a importância de existirem linguagens, ferramentas e bibliotecas de programação capazes de lidar de maneira extremamente eficiente com matrizes desse tamanho. A semântica vetorial tem a vantagem de representar dados por meio de vetores, mas é claro que é preciso um arcabouço de ferramentas tecnológicas capazes de lidar de forma eficiente com esse tipo de estrutura de dados. *Bacana?* Prosseguindo...

Nesse tipo de matriz, a representação de uma palavra pode ser calculada a partir da contagem de sua co-ocorrência com cada palavra do vocabulário em um determinado contexto. *Que contexto seria esse, professor?* Depende! Pode ser um site, um livro, um capítulo, um parágrafo, uma frase qualquer, entre outros. Uma célula da matriz registraria, portanto, quantas vezes duas palavras aparecem dentro de algum desses contextos.

Ocorre que não é muito bacana utilizar contextos muito grandes. *Por que, Diego?* Vamos pensar em um livro bem grande: *O Senhor dos Anéis* (1568 páginas). Nesse livro, é possível encontrar as palavras "*condado*" e "*pernicioso*". *E o que isso me diz?* Nada! Em contextos muito grandes, é difícil encontrar relações relevantes entre palavras. É mais interessante utilizar contextos menores, em geral em torno de uma palavra-alvo.

Por exemplo: nós podemos pegar uma janela de ± 4 palavras – quatro palavras à esquerda de uma palavra-alvo e quatro palavras à direita de uma palavra-alvo. *Por que essa é uma boa estratégia, professor?* Porque é mais provável que palavras mais próximas tenham uma maior relação em um contexto menor do que em um contexto maior, isto é, quanto menor o contexto, maior as chances de palavras próximas terem uma maior relação. Vejam o exemplo da tabela a seguir:

	AARDVARK[10]	...	AÇÚCAR	...	COMPUTADOR	...	DADOS	...	RESULTADO	...	TORTA	...
CEREJA	0	...	25	...	2	...	8	...	9	...	442	...
MORANGO	0	...	19	...	0	...	0	...	1	...	60	...
DIGITAL	0	...	4	...	1670	...	1683	...	85	...	5	...
INFORMAÇÃO	0	...	13	...	3325	...	3982	...	378	...	5	...

Essa tabela representa o corpus da Wikipedia! Como ele é absurdamente gigantesco, eu selecionei algumas palavras por questões didáticas. Vejamos alguns exemplos de frases que poderiam existir:

```
... é tradicionalmente acompanhada de torta de cereja, uma sobremesa comum na ...
... geralmente misturada com o açúcar do morango. Já a torta francesa é mais fácil ...
... computador pessoal e assistente digital. Esses equipamentos eram os dois ...
... um computador. Isso inclui toda informação disponível na internet. Por volta de ...
```

Vocês notaram que o contexto agora é composto apenas das quatro palavras anteriores e as quatro palavras posteriores? Pois é! Os dados da tabela nos mostram que as palavras *cereja* e *morango* são mais similares entre si do que com outras palavras como *digital*; já *digital* e *informação* são mais similares entre si do que com outras palavras como *morango*. Lembrando da hipótese distributiva, vemos que as palavras que aparecem em contextos semelhantes têm significados semelhantes.

E isso é até meio óbvio: as palavras *cereja* e *morango* geralmente aparecem em contextos semelhantes, logo têm significados semelhantes – ambas são frutas vermelhas. Lembrando que algumas palavras podem ser ambíguas, como o exemplo da manga. Por meio dessas técnicas de semelhança entre vetores, computadores podem inferir de qual manga estamos falando ao analisar outros contextos.

Como a hipótese de distribuição afirma que palavras que ocorrem em contextos semelhantes tendem a ter significados semelhantes, a comparação de vetores permite que o computador saiba fazer essa distinção. Por fim, só mais uma coisa: a tabela que nós vimos na página anterior possui números bastante evidentes para que possamos identificar uma relação entre os vetores, mas – quando temos milhares de dimensões – isso se torna inviável.

Não vamos entrar em detalhes, mas saibam que existem técnicas, como a Semelhança por Cossenos, que permite chegar a um valor entre 0 e 1 para indicar quão semelhantes são vetores (quanto mais próximo de 1, mais semelhantes; quanto mais próximo de 0, mais díspares). Bem, pessoal... é isso! A ideia aqui foi mostrar para vocês como representar um texto em linguagem natural (e seus significados) por meio de vetores bidimensionais (ou matrizes). *E por quê?*

Em primeiro lugar, porque vetores são dados estruturados e nós já sabemos que computadores são apaixonados por dados estruturados; em segundo lugar, porque se trata de uma estrutura de dados bastante dinâmica que permite diversos tipos de manipulação; e em terceiro lugar, porque permite

[10] Observação: eu não inventei essa palavra – ela é a primeira em ordem alfabética na Wikipedia e é o nome de uma espécie de porco afriano.

verificar a semelhança entre vetores em um determinado contexto inferindo sobre seu significado. *Fechou?* Agora vamos estudar alguns algoritmos...

Word2Vec

WORD2VEC

Trata-se de uma técnica de aprendizado de máquina que usa redes neurais para aprender as relações entre palavras em um corpus de texto. É usado para gerar representações vetoriais de alta dimensão de palavras que capturam suas propriedades semânticas e sintáticas, podendo ser usado para identificar palavras semelhantes, descobrir as relações entre elas e até gerar novas palavras.

O Word2Vec é o algoritmo base que utiliza uma arquitetura de redes neurais lineares (não profundas) para prever palavras de contexto a partir de palavras-alvo (Modelo Skip-Gram) ou prever uma palavra-alvo a partir de um contexto (Modelo CBoW). Ele basicamente transforma uma palavra em vetor de significados em que as dimensões desse vetor estão fortemente correlacionadas a elementos semânticos. *Entendido?* Vamos ver rapidamente o CBoW...

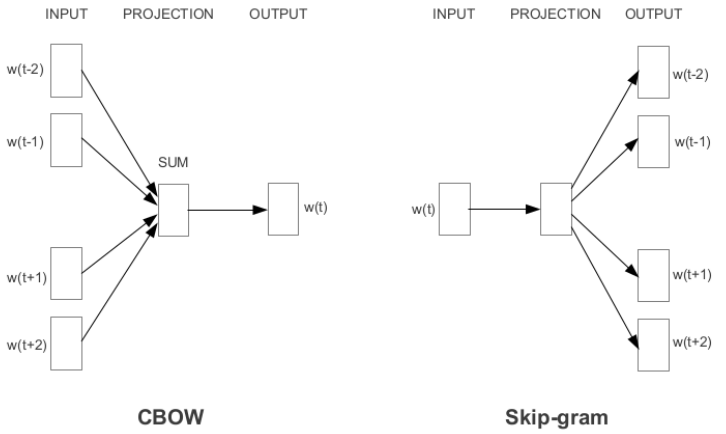
O Continuous Bag-of-Words (CBoW) é um modelo utilizado para descobrir a palavra central de uma sentença, baseado nas palavras que a cercam. Tomemos a frase:

O cachorro _____ atrás do gato

É mais provável que a lacuna seja preenchida com "*correu*" do que "*pedra*". Já o Skip-Gram busca descobrir o contexto baseado na palavra central. Tomemos a palavra:

_____ nave _____

É mais provável que uma possível palavra de contexto seja "*espacial*" do que "*peculato*". Vejam na imagem seguinte redes neurais que representam cada modelo de Word2Vec:



Comparativo: Skip-Gram funciona bem até com poucos dados de treino e representa bem até palavras pouco frequentes; CBoW permite um treinamento muito mais rápido que o Skip-Gram, com acurácia levemente melhor para palavras frequentes. Além disso, o Skip-Gram necessita de uma janela de contexto de cerca de 10 palavras (5 palavras antes da palavra central e 5 palavras depois); já o CBoW necessita de menos...

Bem, na hora da prova será mais difícil memorizar qual é qual! Então eu inventei um mnemônico bobo para ajudá-los a decorar:

Por fim, é importante entender que um *embedding* é basicamente um mapeamento de uma palavra para um espaço vetorial usado para capturar informações semânticas sobre os objetos que representados por essa palavra. Ocorre que é possível ter *word embeddings* pré-treinados, isto é, você passa um volume gigantesco de textos para o algoritmo de forma que ele já armazene o contexto mais provável para cada palavra de um dicionário em um *cache* (área de memória).

Ao armazenar os vetores de um vocabulário em *cache*, aumenta-se a eficiência de acesso e reduzem-se os custos de implantação, dado que já possuímos um dicionário estático. Então imaginem só: alguém coletou uma quantidade colossal de textos e, utilizando um conjunto imenso de máquinas, treinou uma rede neural para cada uma das palavras de um dicionário a fim de prever cada contexto e vice-versa e armazenou em um *cache* de forma estática.

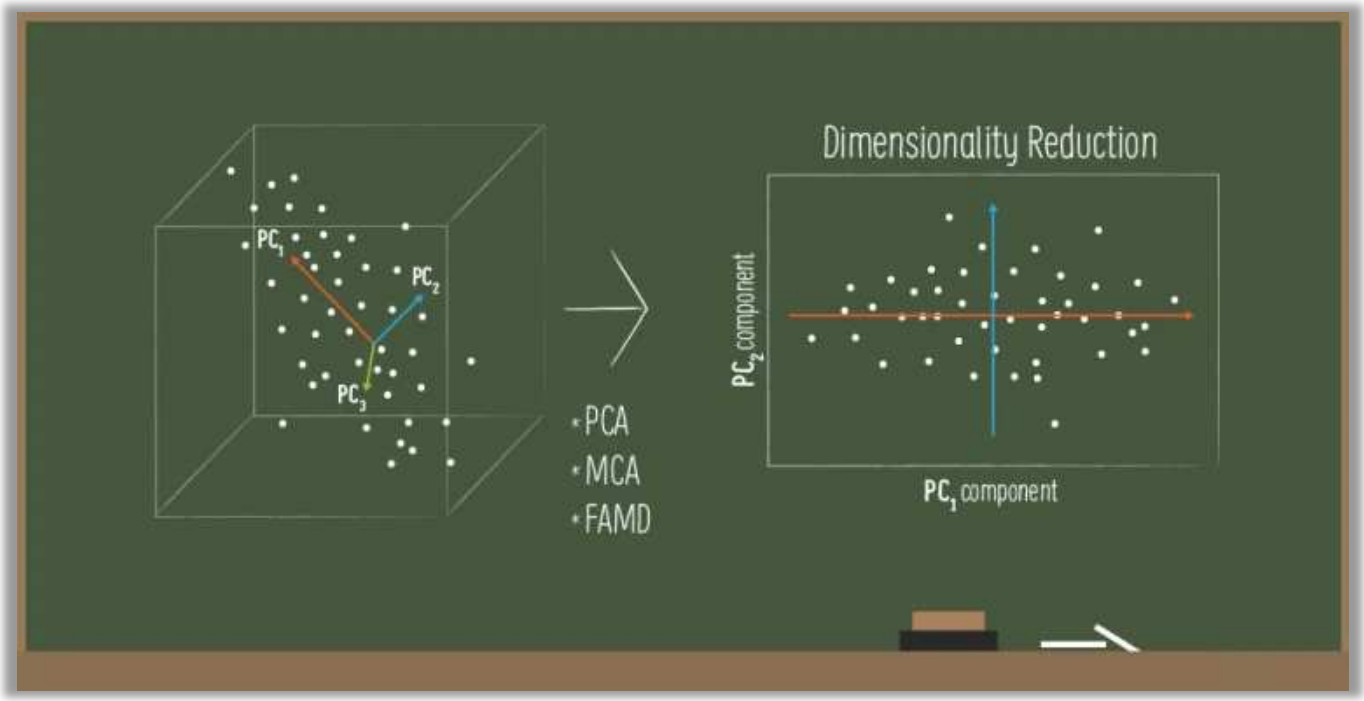
É claro que a maioria desses modelos foram treinados com uma base de dados de palavras em inglês, mas atualmente já fizeram um equivalente em outros idiomas, tal como o português brasileiro. Os principais métodos de *word embedding* pré-treinados são: Word2Vec (2013), GloVe (2014), Wang2Vec (2015) e FastText (2016). E inclusive já as versões em português, tal como a NILC-*Embeddings*.

Redução de Dimensionalidade

RELEVÂNCIA EM PROVA: BAIXÍSSIMA

REDUÇÃO DE DIMENSIONALIDADE

Trata-se do processo de transformar um grande conjunto de dados baseados em texto em um conjunto menor de recursos que capturam as informações mais importantes. Esse processo reduz o número de recursos necessários para representar os dados, facilitando sua análise e interpretação. Também reduz o ruído e ajuda a identificar padrões nos dados.



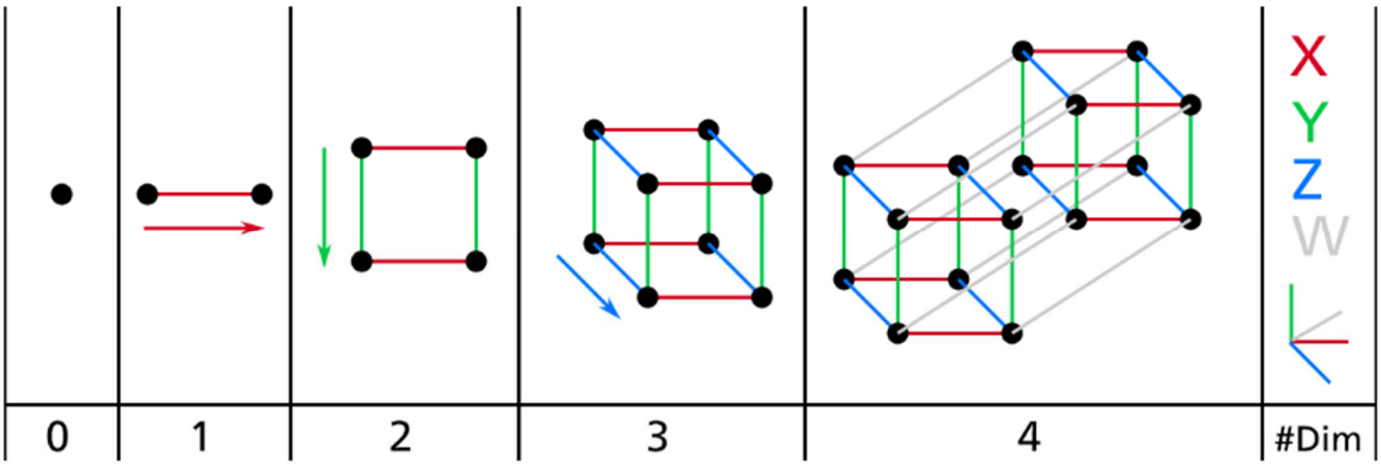
Eu não sei se vocês notaram, mas no tópico anterior eu – muito convenientemente – apresentei uma matriz com poucas palavras (apenas seis colunas e quatro linhas)! Ocorre que geralmente um *corpus* real possui milhares de palavras (coisa de 50.000 palavras). E aí, nós temos um problema sério: quando rodamos esses algoritmos para gerar a tabela, geramos uma matriz com muitas dimensões e a imensa maioria dos valores zerados. *Vocês conseguem imaginar o porquê?*

Ora, porque se o meu contexto envolve uma janela de quatro palavras anteriores ou posteriores, e o meu vocabulário é composto por dezenas de milhares de palavras, nós teremos uma minoria de palavras que possuem uma co-ocorrência e uma imensa maioria de palavras que não possuem qualquer co-ocorrência. Aliás, matrizes que possuem a maioria de valores zerados são conhecidas como matrizes esparsas. Vejam um exemplo um pouquinho mais próximo da realidade...

	PALAVRA 1	PALAVRA 2	PALAVRA 3	PALAVRA 4	PALAVRA 5	PALAVRA 6	PALAVRA 7	PALAVRA 8	PALAVRA 9	PALAVRA 10	PALAVRA 11	PALAVRA 12	PALAVRA 13	PALAVRA 14	PALAVRA 15	PALAVRA 16	PALAVRA 17	...
P01	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	1	0	0
P02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P03	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
P04	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P05	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P06	0	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P07	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
P08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
P09	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
P10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P11	2	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
P12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Esse exemplo ainda não é ideal – eu o coloquei apenas para vocês tenham uma noção um pouquinho melhor do que estamos falando. Note que temos 204 células: 192 zeradas e 12 não zeradas. *E por que isso é um problema, professor?* Porque quanto mais colunas, mais poder computacional precisamos para processar os dados. Além disso, a maioria dos valores não nos acrescenta absolutamente nada justamente porque estão zerados.

Nós estamos tão acostumados a viver em três dimensões que se torna muito difícil imaginar qualquer coisa que tenha mais dimensões. Imaginar uma quarta dimensão já é complexo, já pensaram em 50.000 dimensões. *Agora vocês já imaginaram como é complexo encontrar correlações e similaridades em tantas dimensões?* Eu gosto de fazer a seguinte metáfora: pensem em uma bolinha pintada de vermelho em algum lugar de uma corda de 100m de extensão.



É difícil localizá-la, mas você vai andando e olhando até encontrar. Agora pensem em uma bolinha pintada de vermelho em um pano de 100m². *Complicado, né?* Seria praticamente o tamanho de um campo de futebol, mas com o tempo você vai encontrá-la. Agora pensem em uma bolinha pintada de vermelho em algum lugar de um galpão de 100m³. Puuutz... agora ferrou de vez! Você vai demorar mais ainda a achar a bolinha. E não acabou...

Agora é difícil de imaginar, mas pensem em um espaço de 50.000 dimensões e você tem que achar a maldita bolinha pintada de vermelho. *Puxado, não é?* A ideia aqui é a mesma: quanto maior o número de dimensões, mais difícil de encontrar similaridades. Voltando ao nosso contexto, quanto maior o *corpus*, mais zeros teremos em nossa matriz esparsa, mais inútil serão a imensa maioria dos dados e mais custoso será seu processamento.

Ao utilizar base de dados extensas, ainda mais sendo composta por textos de domínios de conhecimento heterogêneos, é inevitável lidar com vetores de características extremamente longos. Além da elevação da complexidade computacional, o uso de representações vetoriais demasiadamente grandes pode não ser a opção mais adequada. Essa hipótese é confirmada no problema conhecido como Maldição da Dimensionalidade. *Como é isso, Diego?*

A Maldição da Dimensionalidade expressa a existência um número ótimo de características (também chamadas de atributos, colunas ou *features*) que podem ser selecionadas em relação ao tamanho da amostra para maximizar o desempenho do aprendizado. Nesse cenário, torna-se conveniente a aplicação de algum procedimento para redução da base de dados, seja pela seleção de características originais ou através de técnicas de redução da dimensionalidade.

A redução da dimensionalidade tem o objetivo de encontrar representações vetoriais menos complexas, criando novas características sintéticas a partir das originais. Em outras palavras, essa técnica busca transformar espaços de alta dimensionalidade em espaços de baixa dimensionalidade (duas ou três dimensões). Para tal, existe algoritmos como PCA (*Principal Component Analysis*) e t-SNE (*t-Distributed Stochastic Neighbourhood Embedding*).

RESUMO

DEFINIÇÕES DE PROCESSAMENTO DE LINGUAGEM NATURAL
Trata-se da tecnologia que envolve a habilidade de transformar texto ou áudio em informações estruturadas e codificadas, baseado em uma ontologia adequada.
Trata-se da habilidade de um programa de computador de compreender a linguagem humana escrita e falada.
Trata-se da habilidade construir um software capaz de analisar, compreender e gerar linguagens humanas naturalmente, permitindo a comunicação com um computador como se fosse um humano.
Trata-se do campo da Inteligência Artificial que permite aos computadores analisar e compreender a linguagem humana, escrita e falada.
Trata-se da capacidade de construir software que gere e compreenda linguagens naturais para que um usuário possa ter conversas naturais com um computador em vez de por meio de programação.
Trata-se do ramo da inteligência artificial que ajuda os computadores a entender, interpretar e manipular a linguagem humana.
Trata-se da manipulação automática da linguagem natural, como fala e texto por software.
Trata-se de uma subárea da ciência da computação, inteligência artificial e da linguística que estuda os problemas da geração e compreensão automática de linguagens humanas naturais.

PROCESSAMENTO DE LINGUAGEM NATURAL (PLN)

Trata-se de um ramo da inteligência artificial que ajuda os computadores a entender, interpretar e manipular a linguagem humana. O PNL permite que as máquinas leiam e entendam a linguagem humana para interpretar comandos, responder a perguntas e realizar tarefas. Ele é usado em muitas aplicações, como tradução automática, atendimento ao cliente automatizado e assistentes pessoais inteligentes.

PROCESSAMENTO DE LINGUAGEM NATURAL		
SOM	ESTRUTURA	SIGNIFICADO
FONOLOGIA	MORFOLOGIA + SINTAXE	SEMÂNTICA + PRAGMÁTICA

FONOLOGIA	Está relacionada ao reconhecimento de sons que compõem as palavras.
MORFOLOGIA	Reconhece as palavras em termos das unidades primitivas que a compõem.

SINTAXE	Define a estrutura de uma frase, com base na forma como as palavras se relacionam.
SEMÂNTICA	Associa significado a uma estrutura sintática, em termos dos significados das palavras que a compõem.
PRAGMÁTICA	Verifica se o significado associado a uma estrutura sintática é realmente o significado mais apropriado no contexto considerado.

PRÉ-PROCESSAMENTO	Trata-se do estudo de estruturas e formação de palavras, com foco na análise dos componentes individuais das palavras. Nesse contexto, trata-se basicamente da realização da tarefa de tokenização (veremos à frente outro contexto).
ANÁLISE LÉXICA	Busca estudar a morfologia das palavras e recuperar informação que será útil em níveis mais profundos de análise. Para tal, realiza uma decomposição morfológica para identificar classes gramaticais de cada um dos tokens selecionados na atividade anterior.
ANÁLISE SINTÁTICA	A análise sintática é aquela que se preocupa com a estrutura das sentenças em uma gramática formal. Ela permite a extração de frases que transmitem mais significado do que apenas as palavras individuais por si só.
ANÁLISE SEMÂNTICA	A análise semântica trata do significado da sentença.
ANÁLISE PRAGMÁTICA	O componente pragmático, por fim, procura incluir o contexto à análise linguística, a fim de permitir a geração de um significado.

ETAPA	DESCRIÇÃO
CAPTAÇÃO DA VOZ (VOICE PICKUP)	Trata-se da tecnologia que utiliza um microfone para detectar ondas sonoras e convertê-las em sinais elétricos. Essa tecnologia é usada em muitos dispositivos, como telefones, computadores e sistemas de reconhecimento de voz. O microfone capta as ondas sonoras da voz do usuário e as converte em sinais elétricos, que são então processados pelo dispositivo para determinar o que o usuário está dizendo.
RECONHECIMENTO DE FALA (SPEECH RECOGNITION)	Trata-se da tecnologia que permite que um dispositivo reconheça e responda a comandos falados. Essa tecnologia pode ser utilizada para controlar um dispositivo ou aplicativo, transcrever áudio em texto ou entender comandos de linguagem natural. Em outras palavras, podemos dizer que se trata da transcrição da fala no texto correspondente ao que foi dito por um humano. Hoje em dia, essa tecnologia está excepcional! Quem costuma utilizar ferramentas de transcrição – como jornalistas – sabe que a taxa de erros é baixíssima.
ENTENDIMENTO DE LINGUAGEM NATURAL (NATURAL LANGUAGE UNDERSTANDING)	Trata-se da tecnologia que se concentra em permitir que os computadores entendam a fala humana e a linguagem natural. Envolve o desenvolvimento de algoritmos e modelos capazes de interpretar e processar a linguagem falada e extrair informações relevantes do texto em linguagem natural, além de permitir que os computadores entendam o significado por trás das palavras faladas – podendo ser usado para tarefas como atendimento automatizado ao cliente (*os famosos chatbots*).
GERAÇÃO DE LINGUAGEM NATURAL (NATURAL LANGUAGE GENERATION)	Trata-se da tecnologia que permite que os computadores gerem automaticamente uma linguagem natural em texto a partir de dados estruturados. Ele é usado em uma variedade de aplicações, incluindo *chatbots* de atendimento ao cliente, geração automatizada de artigos de notícias e assistentes virtuais. Pode ser usado para gerar resumos, relatórios e outras saídas de texto de fontes de dados estruturadas, como bancos de dados, planilhas e documentos XML.

SÍNTESE DE FALA (SPEECH SYNTHESIS)	Trata-se da tecnologia de produção artificial da fala humana. Por meio da síntese de fala, os computadores são capazes de gerar fala semelhante à humana usando a tecnologia Text-To-Speech (TTS). Essa tecnologia é utilizada em muitos aplicativos, como produtos habilitados para fala, aplicativos de conversão de texto em fala e assistentes virtuais. Hoje em dia, já existem sintetizadores de vozes digitais quase impossíveis de serem identificados como produzidos por uma máquina.

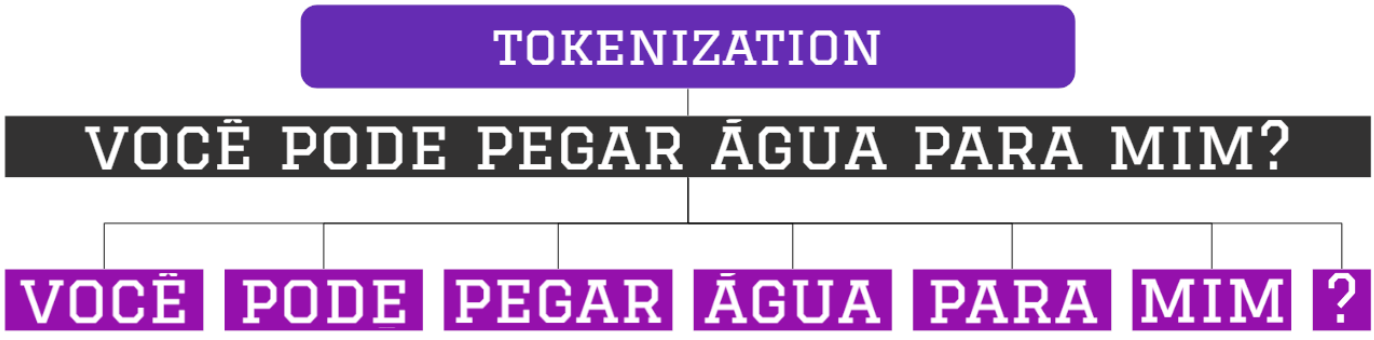
PRÉ-PROCESSAMENTO EM PLN

Trata-se de uma variedade de técnicas usadas para preparar o texto para análise posterior. Isso pode envolver a transformação do texto em uma forma mais receptiva à análise, normalizando o texto, limpando o texto e/ou extraindo recursos do texto. Exemplos de técnicas de pré-processamento incluem tokenização, remoção de stopword, lematização, entre outros.

EXEMPLOS DE TÉCNICAS DE NORMALIZAÇÃO OU LIMPEZA DE DADOS
Remover espaços em branco e pontuação duplicados.
Remover acentuação gráfica, visto que isso ajuda a reduzir erros relacionados ao tipo de codificação de caracteres.
Transformar maiúsculas em minúsculas (exceto quando queremos extrair informações como nomes e locais).
Remover ou substituir caracteres especiais (Ex: &$#@) e emojis (Ex: 😊).
Remover contrações (Ex: caixa d'água).
Transformar numerais das palavras em números (Ex: vinte e três → 23).
Substituir valores pelo seu tipo (Ex: R$50 → Dinheiro; 100Kg → Peso).
Normalizar siglas (Ex: RJ → Rio de Janeiro) e abreviações/vocabulários informais (Ex: pfv → por favor).
Normalizar formatos de data, números de CPF ou outros dados que tenham um formato padrão definido.
Corrigir a ortografia de palavras incorretas.
Remover variações de gênero, tempo, grau e número.
Substituir palavras raras por sinônimos mais comuns.
Remover tags HTML, CSS, JavaScript, etc – além de URLs.
Padronizar de palavras com caracteres minúsculos.

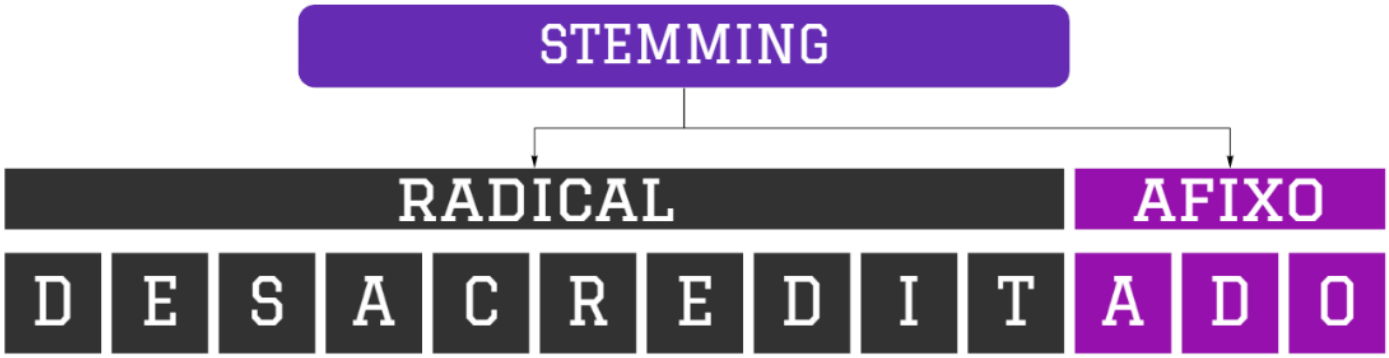
TOKENIZATION

A tokenização, conhecida como segmentação de palavras, é responsável por quebrar a sequência de caracteres em um texto, localizando o limite de cada palavra, isto é, os pontos onde uma palavra termina e outra começa.



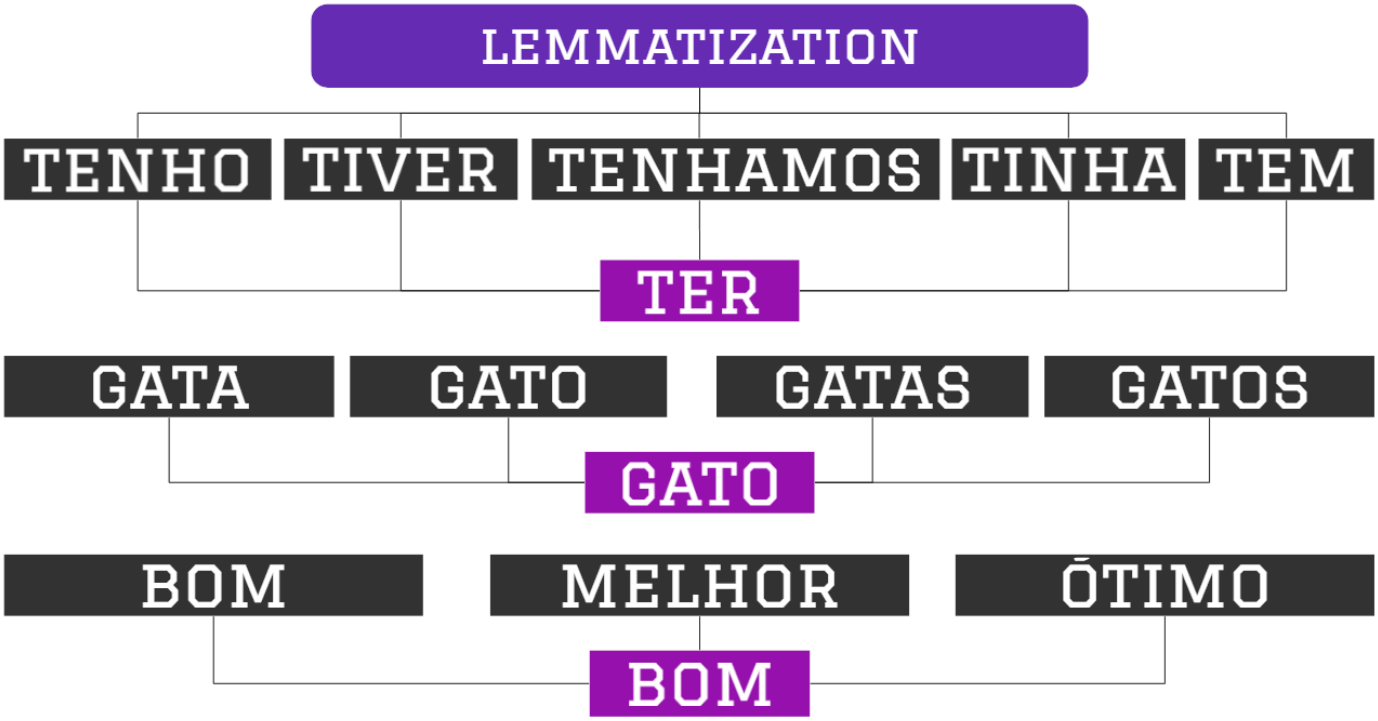
STEMMING

A stemização (do inglês, *stemming*) é o processo que basicamente consiste em reduzir uma palavra ao seu radical – removidos seus afixos (prefixos, infixos e sufixos).



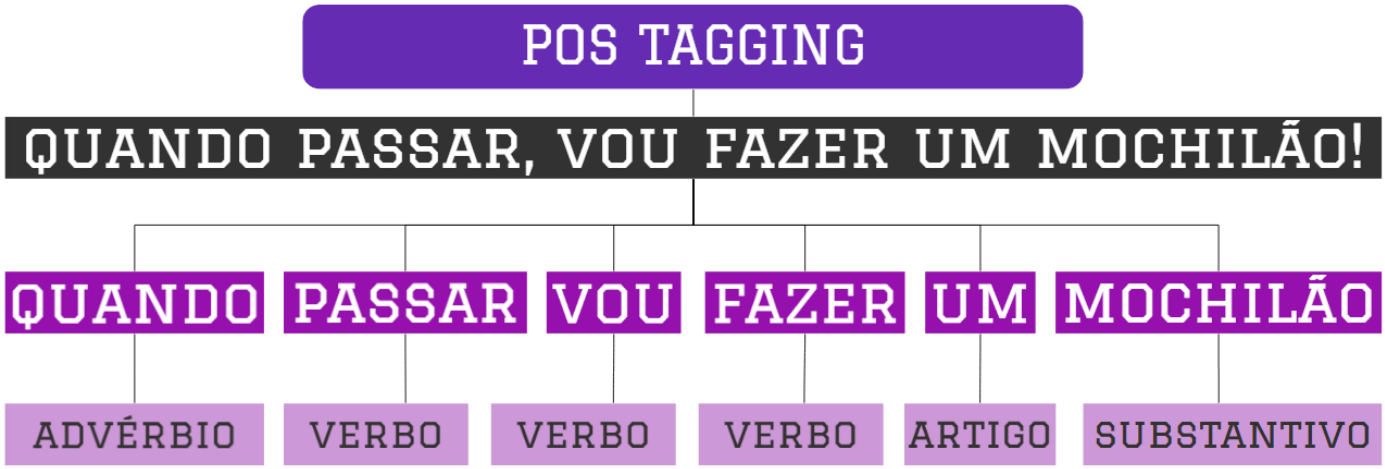
LEMMATIZATION

A lematização (do inglês, *lemmatization*) é o processo de reduzir uma palavra ao seu lema, isto é, à sua forma base, primitiva, nuclear ou canônica – além de agrupar diferentes formas de uma mesma palavra ou seus sinônimos.



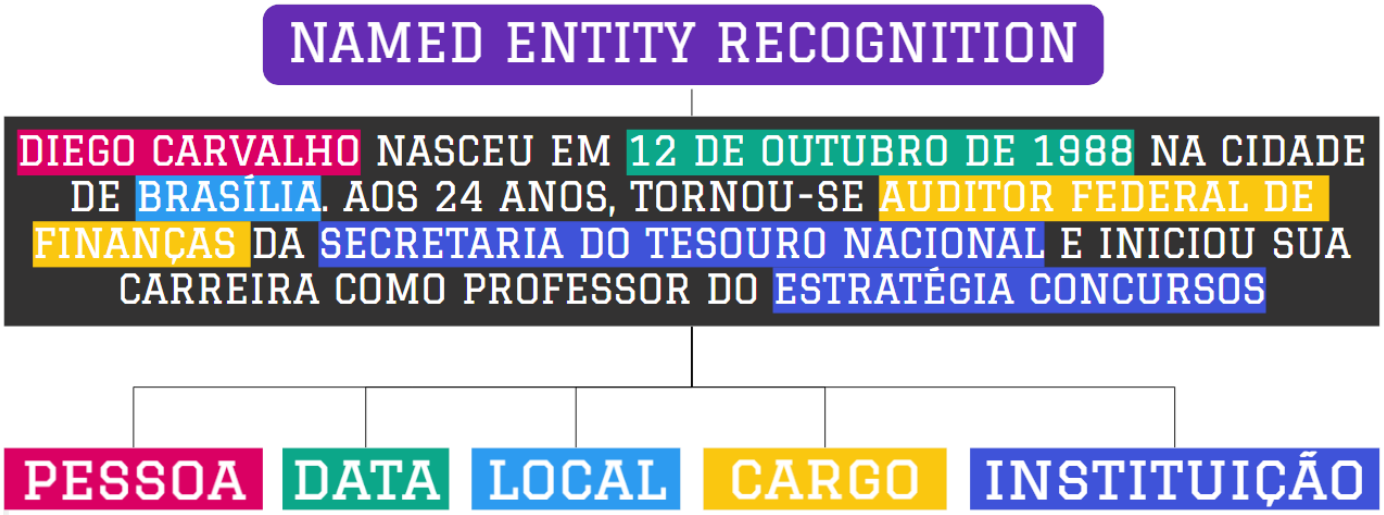
POS-TAGGING

POS Tagging (Part-Of-Speech Tagging) é o processo de identificar classes gramaticais em um texto com o intuito de complementar o processo de extração de palavras relevantes.



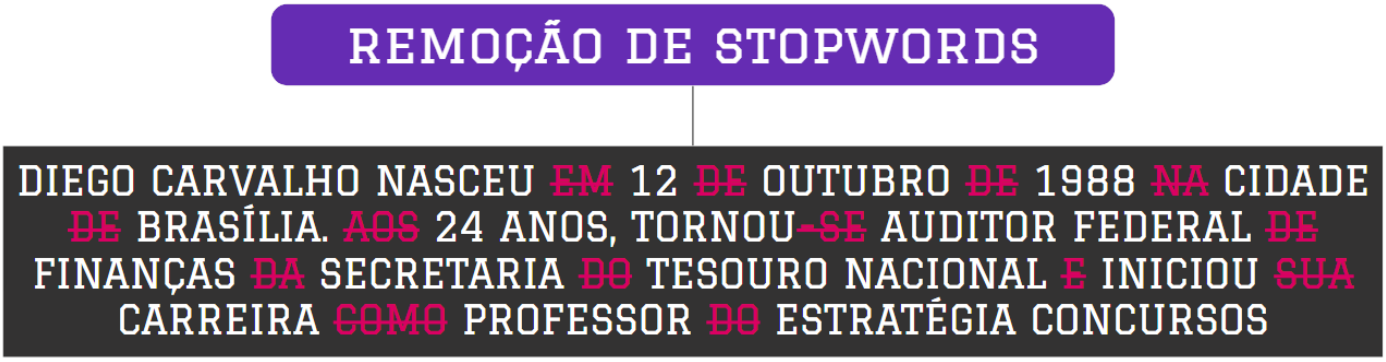
NAMED ENTITY RECOGNITION

O Reconhecimento de Entidades Nomeadas (do inglês, *Named Entity Recognition*) é o processo de reconhecer entidades em um texto, tais como pessoas, datas, organizações, localizações, entre outros.



REMOÇÃO DE STOPWORDS

Stopwords são palavras que podem ser consideradas de pouco valor para o entendimento do sentido de um texto, isto é, palavras semanticamente irrelevantes.



REPRESENTAÇÃO DE TEXTO

Trata-se do processo de transformar texto em estruturas ou representações numéricas que podem ser usadas por algoritmos de aprendizado de máquina para processar e analisar o texto. Isso inclui técnicas como tokenização, lematização, remoção de palavras irrelevantes, entre outros. A representação de texto é uma etapa importante em muitas tarefas de processamento de linguagem natural, como classificação de texto e análise de sentimento.

BAG OF WORDS (SACOLA DE PALAVRAS)

Trata-se de um método de representação de texto como dados numéricos. Envolve atribuir um valor numérico a cada palavra em um documento de texto, com base em sua frequência de ocorrência no documento. Os valores numéricos são usados para criar uma matriz esparsa, que pode ser usada para analisar o documento em busca de padrões ou tendências.

BAG OF WORDS

O CACHORRO SUBIU	VETOR: [1, 1, 1, 0, 0, 0, 0]
O CACHORRO SUBIU NO SOFÁ	VETOR: [1, 1, 1, 1, 1, 0, 0]
O CACHORRO CAIU DO SOFÁ	VETOR: [1, 1, 0, 0, 1, 1, 1]
O SOFÁ CAIU NO CACHORRO	VETOR: [1, 1, 0, 1, 1, 1, 0]

TF-IDF

TF-IDF (Term Frequency–Inverse Document Frequency) é uma estatística numérica usada para medir a relevância de uma palavra para um documento em um corpus. Trata-se de uma combinação de duas métricas: Frequência de Termo (TF) e Inverso da Frequência do Documento (IDF). O primeiro mede a frequência com que uma palavra aparece em um documento, enquanto o segundo mede a importância de uma palavra para um documento em uma coleção. O TF-IDF é usado para medir a relevância de uma palavra para um documento em uma coleção e é comumente usado na recuperação de informações e mineração de texto.

$$\textbf{TFIDF} = \text{TF x IDF}$$

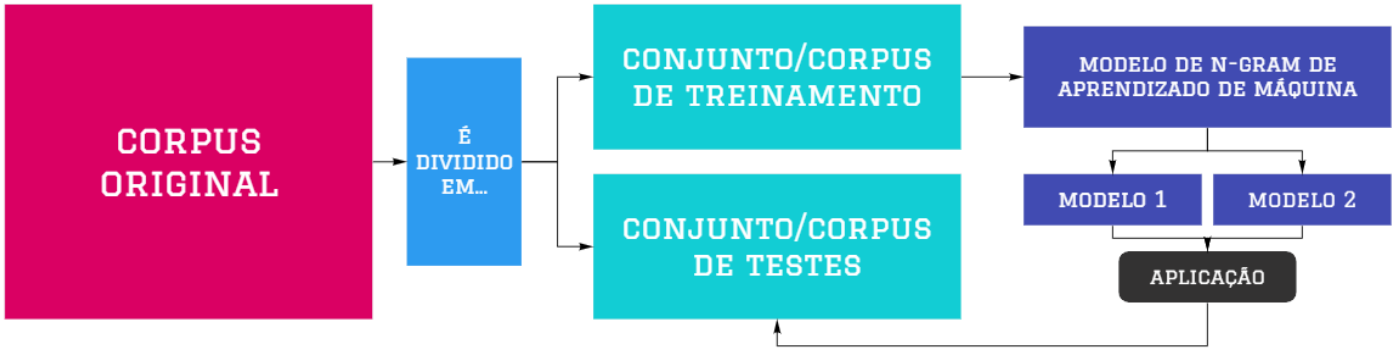
Sendo que:

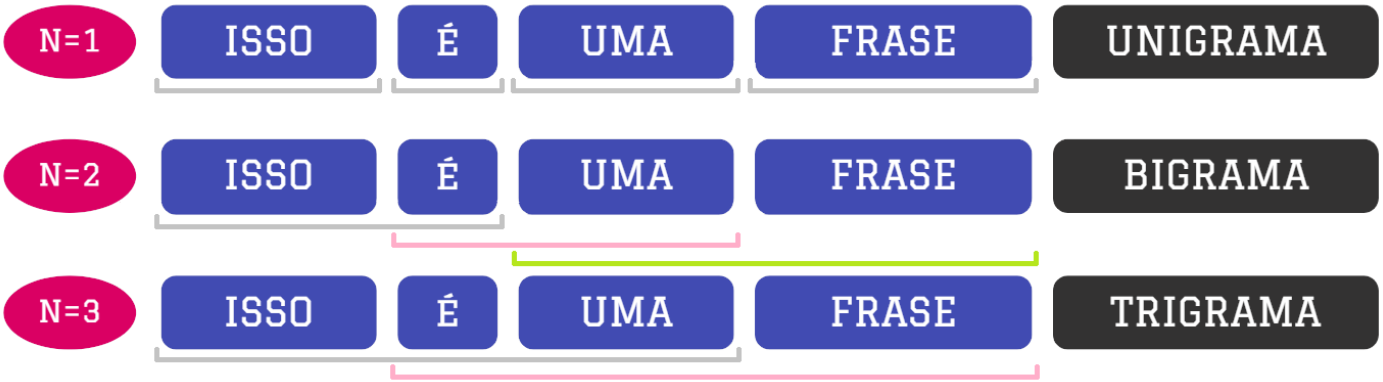
$$TF(t,d) = \frac{\text{QUANTIDADE DE TERMOS } \mathbf{t} \text{ NO DOCUMENTO } \mathbf{d}}{\text{QUANTIDADE DE TERMOS NO DOCUMENTO } \mathbf{d}}$$

$$IDF(N,n) = log \frac{\text{QUANTIDADE (N) TOTAL DE DOCUMENTOS}}{\text{QUANTIDADE (n) DE DOCUMENTOS QUE CONTÊM O TERMO } \mathbf{t}}$$

N-GRAMAS

Trata-se de conjuntos de palavras ou termos adjacentes em um determinado texto ou fala. Eles podem ser usados para analisar o contexto e o significado de um determinado texto e são normalmente usados para criar recursos para algoritmos de aprendizado de máquina, como classificação de texto e análise de sentimento.





CLASSIFICAÇÃO DE TEXTOS

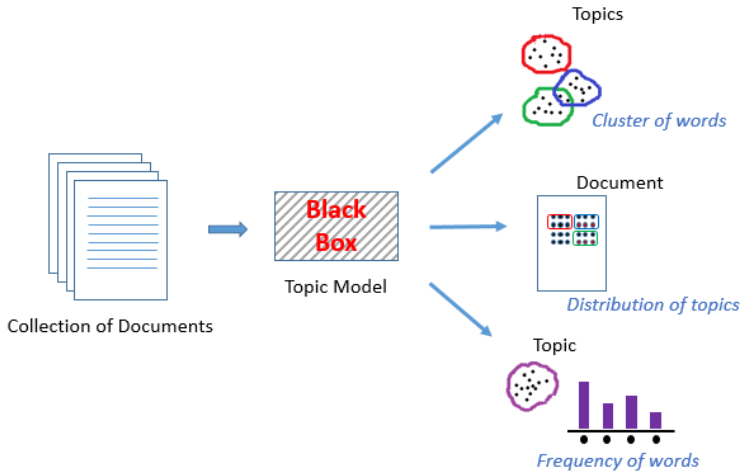
Trata-se do processo de atribuir um determinado documento de texto, como uma frase, parágrafo ou artigo curto, a uma ou mais categorias ou classes predefinidas. Isso é feito usando algoritmos de aprendizado de máquina supervisionados que são treinados em documentos de texto rotulados. O objetivo é determinar a categoria de um determinado documento de texto e usar essa informação para melhorar a precisão das tarefas de processamento de linguagem natural de um sistema. Aplicações comuns de classificação de texto incluem análise de sentimento, detecção de spam e categorização de documentos.

ANÁLISE DE SENTIMENTOS

Também conhecida como Mineração de Opinião, trata-se do processo que busca identificar e extrair opiniões de um texto. Em geral, envolve a classificação de um trecho de texto como sentimento positivo, negativo ou neutro e pode ser usado para detectar opiniões sobre um determinado tópico, produto ou serviço em avaliações online, conversas em redes sociais e outras formas de dados de texto.

MODELAGEM DE TÓPICOS LATENTES

Trata-se de um tipo de aprendizado de máquina não supervisionado utilizado para analisar uma coleção de documentos e descobrir os tópicos subjacentes que estão presentes no texto. O modelo funciona encontrando padrões nas palavras usadas nos documentos e usa esses padrões para agrupar palavras semelhantes em tópicos. A modelagem de tópicos latentes pode ajudar a identificar tópicos em grandes quantidades de texto, como artigos de notícias, postagens de mídia social ou páginas da web, sem intervenção manual. Também pode ser usado para fornecer informações sobre o conteúdo semântico dos documentos.



SEMÂNTICA VETORIAL

Trata-se de um tipo de análise semântica computacional que usa representações baseadas em vetores para palavras. Este método analisa o contexto das palavras em uma frase ou documento para criar um vetor numérico de valores com base na co-ocorrência de palavras nesse contexto. Esse vetor numérico, ou espaço vetorial, é então usado para representar o significado das palavras no texto. A semântica vetorial pode ser usada para executar tarefas como similaridade e classificação de palavras, modelagem de tópicos e análise de sentimentos.

HIPÓTESE DISTRIBUTIVA

QUANTO MAIS SEMANTICAMENTE SEMELHANTES FOREM DUAS PALAVRAS, MAIS SEMELHANTES EM TERMOS DE DISTRIBUIÇÃO ELAS SERÃO E, PORTANTO, MAIS TENDERÃO A OCORRER EM CONTEXTOS LINGUÍSTICOS SEMELHANTES

PALAVRAS QUE APARECEM EM CONTEXTOS SEMELHANTES TENDEM A TER SIGNIFICADOS SEMELHANTES

SEMÂNTICA VETORIAL

DOCUMENTOS SEMELHANTES TENDEM A POSSUIR PALAVRAS SEMELHANTES

DOCUMENTOS QUE POSSUAM PALAVRAS SEMELHANTES TENDEM A POSSUIR VETORES SEMELHANTES

WORD2VEC

Trata-se de uma técnica de aprendizado de máquina que usa redes neurais para aprender as relações entre palavras em um corpus de texto. É usado para gerar representações vetoriais de alta dimensão de palavras que capturam suas propriedades semânticas e sintáticas, podendo ser usado para identificar palavras semelhantes, descobrir as relações entre elas e até gerar novas palavras.

REDUÇÃO DE DIMENSIONALIDADE

Trata-se do processo de transformar um grande conjunto de dados baseados em texto em um conjunto menor de recursos que capturam as informações mais importantes. Esse processo reduz o número de recursos necessários para representar os dados, facilitando sua análise e interpretação. Também reduz o ruído e ajuda a identificar padrões nos dados.

QUESTÕES COMENTADAS – DIVERSAS BANCAS

1. **(CESPE / Petrobrás – 2022)** O CBOW é um modelo de aprendizado de máquina desenhado para prever contexto com base em determinada palavra.

Comentários:

Se eu estou partindo da palavra central para descobrir o contexto, trata-se do Skip-Gram. Lembrem-se do mnemônico: se estou partindo do contexto para descobrir a palavra central, trata-se do CBOW (o que não é o caso).

Gabarito: Errado

---

2. **(CESPE / SEFAZ-SE – 2022)** Na mineração de texto, o processo utilizado para remover os prefixos e sufixos de palavras, de modo a permanecer somente a raiz delas, com a finalidade de melhorar o armazenamento, é conhecido como:

a) *stemming*.
b) análise léxica.
c) remoção de *stop-words*.
d) criação de tesauros.
e) determinação de pesos.

Comentários:

(a) Correto. O processo de extrair prefixos e sufixos de palavras, mantendo apenas a raiz, é conhecido como *stemming*; (b) Errado. Análise léxica é uma das etapas do processo de compilação de linguagens de programação; (c) Errado. Remoção de stop-words é uma das etapas de pré-processamento que busca remover palavras que ocorrem com alta frequência, mas que não acrescentam significado a um texto; (d) Errado. Terauros são recursos que agrupam palavras de acordo com similaridade, isto é, sinônimos; (e) Errado. Determinação de pesos é um processo que ocorre em redes neurais e, não, em mineração de texto.

Gabarito: Letra A

---

3. **(CESPE / PETROBRAS – 2022)** *Stop-words* constituem um conjunto de palavras que proporcionam pouca informação para o significado de uma frase.

Comentários:

Perfeito! As stopwords são palavras muito frequentes no texto, mas que não possuem grande relevância, geralmente são artigos masculinos, femininos, preposições, dentre outros, mas se faz

necessária análise para garantir que a retirada destas stopwords não vá deturpar a compreensão do texto mais relevante.

Gabarito: Correto

---

4. **(CESPE / PETROBRAS – 2022)** Suponha que a palavra **amor** ocorra 1.000 vezes no último livro escrito por certo autor, que escreveu, no total, 10 livros. Nesse caso, se a palavra **amor** for encontrada em todos os livros desse autor, então o valor do TF-IDF (*Term Frequency-Inverse Document Frequency*) referente à palavra **amor** no último livro escrito será igual a 1/1.000.

Comentários:

Vamos analisar ponto por ponto: (1) a palavra analisada é "amor"; (2) essa palavra ocorreu 1.000 vezes no último livro; (3) o autor escreveu um total de 10 livros; (4) considera-se que essa palavra também ocorreu em todos os livros do autor. Vamos relembrar algumas coisas...

O TF-IDF é alto quando a frequência de um termo é elevada dentro de um documento específico e baixa dentro de outros documentos. Lembrando que a fórmula é:

$$\textbf{TFIDF} = \text{TF x IDF}$$

Sendo que:

$$\boldsymbol{TF(t,d)} = \frac{\text{QUANTIDADE DE TERMOS } \mathbf{t} \text{ NO DOCUMENTO } \mathbf{d}}{\text{QUANTIDADE DE TERMOS NO DOCUMENTO } \mathbf{d}}$$

$$\boldsymbol{IDF(N,n)} = log\ \frac{\text{QUANTIDADE (N) TOTAL DE DOCUMENTOS}}{\text{QUANTIDADE (n) DE DOCUMENTOS QUE CONTÊM O TERMO } \mathbf{t}}$$

Sabemos que t = ***amor*** e d = ***livro***. Logo, temos que:

$$\boldsymbol{TF(amor,livro)} = \frac{\text{QUANTIDADE DE TERMOS } \mathbf{amor} \text{ NO DOCUMENTO } \mathbf{livro}}{\text{QUANTIDADE DE TERMOS NO DOCUMENTO } \mathbf{livro}} = 1000/?$$

$$\boldsymbol{IDF(10,10)} = log\ \frac{\text{QUANTIDADE (N) TOTAL DE DOCUMENTOS}}{\text{QUANTIDADE (n) DE DOCUMENTOS QUE CONTÊM O TERMO } \mathbf{t}} = log\frac{10}{10} = \log 1 = 0$$

Note que não sabemos a quantidade de termos do último livro, mas não é necessário porque sabemos que TF-IDF = TF x IDF. E IDF = 0, logo TF-IDF = TF x 0 = 0 e, não, 1/1000.

Gabarito: Errado

---

5. **(CESPE / SEFAZ-CE – 2021)** Um dos desafios do processamento de linguagem natural (PLN) é a polissemia, ou seja, a característica de palavras e frases poderem ter mais de um significado.

Comentários:

Perfeito! A polissemia é o fenômeno que ocorre quando uma palavra ou frase possuem mais de um significado.

Gabarito: Correto

---

6. **(CESPE / SEFAZ-CE – 2021)** Aplicações de reconhecimento de voz fazem a transcrição de um áudio para texto diretamente, sem a necessidade de nenhum modelo intermediário.

Comentários:

Apesar de a questão ser de 2021, trata-se de um contexto anacrônico! Há alguns anos, era necessário ter um modelo intermediário que realizava uma modelagem acústica/fonética com reconhecimento de fonemas usando Modelo Hidden Markov para identificar palavras correspondentes mais prováveis. No entanto, desde 2014 já existem sistemas ponto a ponto em redes neurais profundas que aprendem conjuntamente todas as etapas de um reconhecimento de fala sem a necessidade de um modelo intermediário. No entanto, o gabarito definitivo foi errado.

Gabarito: Errado

---

7. **(CESPE / MEC – 2015)** O processo de aplicação de operações em uma palavra, a fim de que seja encontrada a etimologia dessa palavra, denomina-se stemming.

Comentários:

Etimologia é o estudo gramatical da origem e história das palavras, de onde surgiram e como evoluíram ao longo dos anos; já o *stemming* busca reduzir uma palavra ao seu radical.

Gabarito: Errado

---

8. **(CESPE / MEC – 2015)** Stop words integram uma lista universal de palavras utilizadas para identificar as paradas ou finais de textos, de modo a auxiliar na análise semântica.

Comentários:

Não, isso não faz o menor sentido! Stopwords são palavras que podem ser consideradas de pouco valor para o entendimento do sentido de um texto, isto é, palavras semanticamente irrelevantes.

Gabarito: Errado

---

9. **(CESPE / ANATEL – 2014)** A tecnologia de análise de sentimento social é um intrincado algoritmo que analisa reações em torno de um tema, marca ou pessoa, sem a necessidade de

uma hashtag. Com imensa capacidade de processamento em tempo real, o sistema consegue identificar, filtrar e analisar os textos em português contidos nos comentários das mídias sociais acerca de determinado tema.

Comentários:

Perfeito! Análise de Sentimento realmente analisa reações sobre um tema, marca ou pessoa. De fato, essa tecnologia não necessita de hashtag para fazer a análise de sentimento. Ela, de fato, consegue identificar, filtrar e analisar os textos em português contidos em redes sociais (ou em qualquer outro conjunto de textos).

Gabarito: Correto

---

10. (FCC / TRF4 – 2019) Um Analista necessita desenvolver uma aplicação chatbot que simula um ser humano na conversação com as pessoas. Para isso o Analista deve usar pesquisa em Processamento de Linguagem Natural – PLN que envolve três aspectos da comunicação, quais sejam,

a) Som, ligado à fonologia, Estrutura que consiste em análises morfológica e sintática e Significado que consiste em análises semântica e pragmática.

b) Áudio, ligado à fonologia, Estrutura que consiste em análises de línguas estrangeiras e Significado que consiste em análises semântica e pragmática.

c) Conversação, ligado à tecnologia de chatbot, Semântica que consiste em análises de línguas estrangeiras e Arquitetura Spelling que realiza as análises sintática e pragmática.

d) Business Intelligence, ligado à tecnologia OLAP, Mining que consiste em análises de línguas em geral e Spelling que realiza as funções de chatbot.

e) Business Intelligence, ligado à tecnologia OLAP, Mining que consiste em análises de línguas em geral e Spelling que realiza as funções de chatbot.

Comentários:

Trata-se de som (fonologia), estrutura (morfologia + sintaxe) e significado (semântica).

Gabarito: Letra A

---

11. (FCC / TRE-RN – 2011 – Letra C) BOW (Bag of Words) é o processo em que substantivos, adjetivos, pronomes e verbos são reduzidos aos seus radicais.

Comentários:

Opa... isso é o Stemming e, não, Bag of Words!

Gabarito: Errado

---

12. (FCC / TRE-RN – 2011 – Letra D) Stop Words é uma matriz onde cada linha representa um documento e cada coluna representa um termo.

Comentários:

Opa... isso seria uma matriz termo-documento! Stopwords são palavras que podem ser consideradas de pouco valor para o entendimento do sentido de um texto, isto é, palavras semanticamente irrelevantes. Em geral, trata-se de artigos, preposições, pronomes e conjunções (Ex: as, e, que, os, de, para, com, sem, aquele, etc).

Gabarito: Errado

---

13. (FCC / TRE-RN – 2011 – Letra E) Leitura, extração, contagem e cálculo de frequência dos termos, são etapas típicas do método *stemming*.

Comentários:

Na verdade, a questão trata do TF-IDF e, não, de *stemming*.

Gabarito: Errado

---

14. (FGV / TCU – 2022) Uma organização está implementando um sistema de busca de informações interno, e a equipe de desenvolvimento resolveu avaliar diferentes modelos de linguagem vetoriais que ajudariam a conectar melhor documentos e consultas em departamentos que usam terminologias distintas em áreas de negócio que se sobrepõem. Um dos analistas ressaltou que seria interessante guardar os vetores de todo o vocabulário do modelo em um cache, de forma a aumentar a eficiência de acesso e reduzir certos custos de implantação.

Das alternativas abaixo, aquela que lista apenas os modelos compatíveis com essa estratégia de *caching* é:

a) TF-IDF, BERT;
b) Word2Vec, BERT, GPT-2;

c) GloVe, GPT-2;
d) Word2Vec, GloVe;
e) GPT-2, BERT.

Comentários:

(1) Word2Vec e GloVe realmente utilizam uma estratégia de *caching*; **(2)** BERT e GPT2 são modelos sequenciais que utilizam Transformers para adaptar a representação vetorial das palavras pelo contexto de outras palavras no qual se encontra. Assim, esses vetores para palavras não são constantes; **(3)** TF-IDF não utiliza representação vetorial para palavras e, sim, para textos.

Gabarito: Letra D

---

15. (FGV / TJDFT – 2022) Considere a sentença a seguir.

s: "O acesso ao auditório também pode ser feito através de uma rampa"

Aplicando a função ***f*** à sentença, obtém-se o seguinte resultado:

***f*(s)** = "acesso auditório pode ser feito através rampa"

A melhor descrição para a tarefa realizada pela função ***f*** é:

a) filtragem de conectivos;
b) lematização;
c) sumarização de sentença;
d) filtragem de stop words;
e) remoção de ruído.

Comentários:

Claramente ocorre uma filtragem de stopwords. Lembrando que stopwords são palavras que podem ser consideradas de pouco valor para o entendimento do sentido de um texto, isto é, palavras semanticamente irrelevantes. Em geral, trata-se de artigos, preposições, pronomes e conjunções (Ex: as, e, que, os, de, para, com, sem, aquele, etc). Essas palavras podem ser ignoradas com segurança, realizando uma pesquisa em uma lista predefinida de palavras-chave, reduzindo o ruído e melhorando o desempenho.

Gabarito: Letra D

---

16. (FGV/ TCU – 2022) Um analista do TCU gostaria de aplicar um modelo de Latent Dirichlet Allocation (LDA) em um conjunto de textos. A alternativa que melhor descreve o resultado do modelo é:

a) uma lista de tópicos, cada um com um título diferente;
b) uma lista das palavras mais importantes no conjunto de documentos;
c) cada documento é classificado em somente um tópico, onde cada tópico é formado por uma lista de palavras;
d) cada documento possui uma distribuição de probabilidade de pertencer a algum dos tópicos, onde cada tópico é formado por uma lista de palavras e cada palavra pertence a somente um tópico;
e) cada documento possui uma distribuição de probabilidade de pertencer a algum dos tópicos, onde cada tópico é formado por uma distribuição de probabilidade sobre todas as palavras presentes nos documentos.

Comentários:

(a) Errado, tópicos não possuem necessariamente um título próprio – e se tiver, ele é atribuído por um especialista e nem sequer precisa necessariamente ter relação com o tópico; (b) Errado, não há nenhuma relação com a importância das palavras; (c) Errado, cada documento pode ser associado a um conjunto de tópicos; (d) Errado, palavras podem pertencer a mais de um tópico; (e) Correto.

A Alocação de Dirichlet Latente é um modelo estatístico generativo que permite que conjuntos de observações sejam explicados por grupos não observados que explicam por que algumas partes dos dados são semelhantes. Por exemplo: se as observações são palavras coletadas em documentos, isso pressupõe que cada documento é uma mistura de um pequeno número de tópicos e que a presença de cada palavra pode ser atribuída a um dos tópicos do documento.

Gabarito: Letra E

---

17. (FGV/ TCU – 2022) Considere os documentos A e B a seguir.

A = "Há pessoas que choram por saber que as rosas têm espinho"
B = "Há outras que sorriem por saber que os espinhos têm rosas"

A submatriz da matriz de TF-IDF desses dois documentos correspondente aos termos "Rosas", "Choram" e "Sorriem", nessa ordem, é:

a) $\begin{bmatrix} 0 & 0 & \frac{1}{11} \\ 0 & \frac{\log 2}{11} & 0 \end{bmatrix}$;

b) $\begin{bmatrix} \frac{1}{11} & \frac{1}{11} & 0 \\ \frac{1}{11} & 0 & \frac{1}{11} \end{bmatrix}$;

c) $\begin{bmatrix} 0 & \frac{\log 2}{11} & 0 \\ 0 & 0 & \frac{\log 2}{11} \end{bmatrix}$;

d) $\begin{bmatrix} 0 & 0 & \frac{\log 2}{11} \\ 0 & \frac{1}{11} & 0 \end{bmatrix}$;

e) $\begin{bmatrix} \frac{1}{11} & \frac{\log 2}{11} & 0 \\ \frac{1}{11} & 0 & \frac{\log 2}{11} \end{bmatrix}$.

Comentários:

Vamos lá! TF (Term Frequency) se refere à frequência de cada palavra em um texto; IDF (Inverse Document Frequency) se refere ao inverso da frequência de uma palavra em um texto. Como em um texto frequentemente temos muitas stopwords (conectores que não possui um valor significativo para o texto), o IDF é utilizado para reduzir o valor numérico dessas palavras – destacando as palavras mais relevantes. Vamos calcular o TF:

	ROSAS	CHORAM	SORRIEM
DOCUMENTO A	1/11	1/11	0/11
DOCUMENTO B	1/11	0/11	1/11

Percebam que a palavra "Rosas" aparece 1 vez dentre as 11 palavras o Documento A e do Documento B; já a palavra "Choram" aparece 1 vez dentre as 11 palavras do Documento A e não aparece no Documento B; por fim, a palavra "Sorriem" aparece 1 vez dentre as 11 palavras do Documento B e não aparece no Documento A. Agora vamos calcular IDF:

	ROSAS	CHORAM	SORRIEM
DOCUMENTO A	log 2/2 = 0	log 2/1 = log 2	log 2/1 = log 2
DOCUMENTO B	log 2/2 = 0	log 2/1 = log 2	log 2/1 = log 2

Percebam que o IDF utiliza o log(N/n), sendo que N representa a quantidade total de documentos; e n representa a quantidade de documentos em que uma determinada palavra aparece. Temos 2 documentos e as três palavras aparecem respectivamente em 2, 1 e 1 documentos. Pronto, agora basta multiplicar o TF pelo IDF → TF x IDF. Dessa forma, teremos:

$$\begin{matrix} 1/11 & 1/11 & 0/11 \\ 1/11 & 0/11 & 1/11 \end{matrix} \times \begin{matrix} 0 & \log 2 & \log 2 \\ 0 & \log 2 & \log 2 \end{matrix} = \begin{matrix} \mathbf{0} & \mathbf{\log 2/11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\log 2/11} \end{matrix}$$

Gabarito: Letra C

LISTA DE QUESTÕES – DIVERSAS BANCAS

1. **(CESPE / Petrobrás – 2022)** O CBOW é um modelo de aprendizado de máquina desenhado para prever contexto com base em determinada palavra.

2. **(CESPE / SEFAZ-SE – 2022)** Na mineração de texto, o processo utilizado para remover os prefixos e sufixos de palavras, de modo a permanecer somente a raiz delas, com a finalidade de melhorar o armazenamento, é conhecido como:

   a) *stemming*.
   b) análise léxica.
   c) remoção de *stop-words*.
   d) criação de tesauros.
   e) determinação de pesos.

3. **(CESPE / PETROBRAS – 2022)** *Stop-words* constituem um conjunto de palavras que proporcionam pouca informação para o significado de uma frase.

4. **(CESPE / PETROBRAS – 2022)** Suponha que a palavra **amor** ocorra 1.000 vezes no último livro escrito por certo autor, que escreveu, no total, 10 livros. Nesse caso, se a palavra **amor** for encontrada em todos os livros desse autor, então o valor do TF-IDF (*Term Frequency-Inverse Document Frequency*) referente à palavra **amor** no último livro escrito será igual a 1/1.000.

5. **(CESPE / SEFAZ-CE – 2021)** Um dos desafios do processamento de linguagem natural (PLN) é a polissemia, ou seja, a característica de palavras e frases poderem ter mais de um significado.

6. **(CESPE / SEFAZ-CE – 2021)** Aplicações de reconhecimento de voz fazem a transcrição de um áudio para texto diretamente, sem a necessidade de nenhum modelo intermediário.

7. **(CESPE / MEC – 2015)** O processo de aplicação de operações em uma palavra, a fim de que seja encontrada a etimologia dessa palavra, denomina-se stemming.

8. **(CESPE / MEC – 2015)** Stop words integram uma lista universal de palavras utilizadas para identificar as paradas ou finais de textos, de modo a auxiliar na análise semântica.

9. **(CESPE / ANATEL – 2014)** A tecnologia de análise de sentimento social é um intrincado algoritmo que analisa reações em torno de um tema, marca ou pessoa, sem a necessidade de uma hashtag. Com imensa capacidade de processamento em tempo real, o sistema consegue identificar, filtrar e analisar os textos em português contidos nos comentários das mídias sociais acerca de determinado tema.

10. **(FCC / TRF4 – 2019)** Um Analista necessita desenvolver uma aplicação chatbot que simula um ser humano na conversação com as pessoas. Para isso o Analista deve usar pesquisa em

Processamento de Linguagem Natural – PLN que envolve três aspectos da comunicação, quais sejam,

a) Som, ligado à fonologia, Estrutura que consiste em análises morfológica e sintática e Significado que consiste em análises semântica e pragmática.

b) Áudio, ligado à fonologia, Estrutura que consiste em análises de línguas estrangeiras e Significado que consiste em análises semântica e pragmática.

c) Conversação, ligado à tecnologia de chatbot, Semântica que consiste em análises de línguas estrangeiras e Arquitetura Spelling que realiza as análises sintática e pragmática.

d) Business Intelligence, ligado à tecnologia OLAP, Mining que consiste em análises de línguas em geral e Spelling que realiza as funções de chatbot.

e) Business Intelligence, ligado à tecnologia OLAP, Mining que consiste em análises de línguas em geral e Spelling que realiza as funções de chatbot.

11. **(FCC / TRE-RN – 2011 – Letra C)** BOW (Bag of Words) é o processo em que substantivos, adjetivos, pronomes e verbos são reduzidos aos seus radicais.

12. **(FCC / TRE-RN – 2011 – Letra D)** Stop Words é uma matriz onde cada linha representa um documento e cada coluna representa um termo.

13. **(FCC / TRE-RN – 2011 – Letra E)** Leitura, extração, contagem e cálculo de frequência dos termos, são etapas típicas do método *stemming*.

14. **(FGV / TCU – 2022)** Uma organização está implementando um sistema de busca de informações interno, e a equipe de desenvolvimento resolveu avaliar diferentes modelos de linguagem vetoriais que ajudariam a conectar melhor documentos e consultas em departamentos que usam terminologias distintas em áreas de negócio que se sobrepõem. Um dos analistas ressaltou que seria interessante guardar os vetores de todo o vocabulário do modelo em um cache, de forma a aumentar a eficiência de acesso e reduzir certos custos de implantação.

Das alternativas abaixo, aquela que lista apenas os modelos compatíveis com essa estratégia de *caching* é:

a) TF-IDF, BERT;
b) Word2Vec, BERT, GPT-2;
c) GloVe, GPT-2;
d) Word2Vec, GloVe;
e) GPT-2, BERT.

15. **(FGV / TJDFT – 2022)** Considere a sentença a seguir.

s: "O acesso ao auditório também pode ser feito através de uma rampa"

Aplicando a função ***f*** à sentença, obtém-se o seguinte resultado:

***f*(s)** = "acesso auditório pode ser feito através rampa"

A melhor descrição para a tarefa realizada pela função ***f*** é:

a) filtragem de conectivos;
b) lematização;
c) sumarização de sentença;
d) filtragem de stop words;
e) remoção de ruído.

16. **(FGV/ TCU – 2022)** Um analista do TCU gostaria de aplicar um modelo de Latent Dirichlet Allocation (LDA) em um conjunto de textos. A alternativa que melhor descreve o resultado do modelo é:

a) uma lista de tópicos, cada um com um título diferente;
b) uma lista das palavras mais importantes no conjunto de documentos;
c) cada documento é classificado em somente um tópico, onde cada tópico é formado por uma lista de palavras;
d) cada documento possui uma distribuição de probabilidade de pertencer a algum dos tópicos, onde cada tópico é formado por uma lista de palavras e cada palavra pertence a somente um tópico;
e) cada documento possui uma distribuição de probabilidade de pertencer a algum dos tópicos, onde cada tópico é formado por uma distribuição de probabilidade sobre todas as palavras presentes nos documentos.

17. **(FGV/ TCU – 2022)** Considere os documentos A e B a seguir.

A = "Há pessoas que choram por saber que as rosas têm espinho"
B = "Há outras que sorriem por saber que os espinhos têm rosas"

A submatriz da matriz de TF-IDF desses dois documentos correspondente aos termos "Rosas", "Choram" e "Sorriem", nessa ordem, é:

a) $\begin{bmatrix} 0 & 0 & \frac{1}{11} \\ 0 & \frac{\log 2}{11} & 0 \end{bmatrix}$;

b) $\begin{bmatrix} \frac{1}{11} & \frac{1}{11} & 0 \\ \frac{1}{11} & 0 & \frac{1}{11} \end{bmatrix}$;

c) $\begin{bmatrix} 0 & \frac{\log 2}{11} & 0 \\ 0 & 0 & \frac{\log 2}{11} \end{bmatrix}$;

d) $\begin{bmatrix} 0 & 0 & \frac{\log 2}{11} \\ 0 & \frac{1}{11} & 0 \end{bmatrix}$;

e) $\begin{bmatrix} \frac{1}{11} & \frac{\log 2}{11} & 0 \\ \frac{1}{11} & 0 & \frac{\log 2}{11} \end{bmatrix}$.

GABARITO – DIVERSAS BANCAS

1. ERRADO
2. LETRA A
3. CORRETO
4. ERRADO
5. CORRETO
6. ERRADO
7. ERRADO
8. ERRADO
9. CORRETO
10. LETRA A
11. ERRADO
12. ERRADO
13. ERRADO
14. LETRA D
15. LETRA D
16. LETRA E
17. LETRA C

ESSA LEI TODO MUNDO CONHECE:
PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.

1. 

Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.

2. 

Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).

3. 

Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).

4. 

Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).

5. 

Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).

6. 

Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).

7. 

Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.

8. 

O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.

Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.