NATURAL LANGUAGE PROCESSING

AULA 4

Prof. Luciano Frontino de Medeiros

CONVERSA INICIAL

O objetivo desta aula é apresentar os modelos de recuperação de documentos e sua relação com o modelo TF-IDF, utilizando-se exemplos dos modelos vetorial e probabilístico. As métricas de comparação de palavras para contemplar as comparações aproximadas também são abordadas, em particular a distância de Levenshtein e as métricas de similaridade de Jaro e Jaro-Winkler. Outro tema a ser estudado é a tradução automática, uma tarefa por excelência da área de NLP. Por fim, a análise de sentimento e mineração de opinião será estudada, mostrando-se então os desafios relacionados com as aplicações que visam identificar os sentimentos do usuário embutidos em documentos ou sentenças.

TEMA 1 – MODELOS DE RECUPERAÇÃO DE DOCUMENTOS E TF-IDF

1.1 Modelo Vetorial

Como estudado anteriormente nos modelos de recuperação de documentos, o modelo booleano caracteriza-se por adotar valores binários (0 ou 1) para os pesos dos termos de indexação e dessa forma é limitado para uma descrição mais precisa. O modelo vetorial lida com valores de pesos não binários, de maneira que se torna possível calcular **graus de similaridade** entre os documentos armazenados e da própria consulta do usuário. Pela ordenação decrescente dos documentos deste grau de similaridade, o modelo vetorial considera documentos que podem casar parcialmente com os termos de consulta. Assim, os documentos são ranqueados de uma maneira mais precisa (Baeza-Yates; Ribeiro-Neto, 2011, p. 77).

Para ilustrar a aplicação do modelo vetorial de busca de documentos, será utilizado o terceiro esquema, a seguir, que utiliza as seguintes fórmulas:

$$(1 + \log f_{ij}) \times \log \frac{N}{n_i} \quad (1)$$

$$(1 + \log f_{iq}) \times \log \frac{N}{n_i} \quad (2)$$

A fórmula (1) foi utilizada para calcular os pesos TF-IDF de conteúdos anteriores. A proposta agora é utilizar a fórmula (2) para demonstrar como uma

consulta de termos permite o ranqueamento por graus de similaridade, conforme o modelo vetorial.

O cálculo utiliza a fórmula do modelo vetorial utilizada em conteúdos anteriores, sendo colocada novamente a seguir para facilitar a demonstração do exemplo:

$$sim(d_j, q) = \frac{\vec{d_j}.\vec{q}}{|\vec{d_j}| \times |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \times \sqrt{\sum_{i=1}^{t} w_{iq}^2}} \quad (3)$$

O grau de similaridade será calculado, portanto, com base no produto interno dividido pela norma. O produto interno usa o valor TF-IDF do termo de busca pelo valor TF-IDF do termo de indexação, somando-se para todos os documentos na coleção considerada. Como no termo de busca, a frequência do termo é igual a 1 (o termo é colocado apenas uma vez na expressão de busca), o valor TF é igual a 1; de forma prática, é como considerar apenas o valor IDF da consulta, pois na fórmula (2) o valor $\log_2 f_{iq} = \log_2 1 = 0$. Na Figura 1 é mostrado o cálculo da TF-IDF e os valores IDF.

Figura 1 - Valores TF-IDF e IDF do exemplo

No.	Termo	d_1	d_2	d_3	d_4	n_i	idf_i
1	atmosfera	-	-	-	2	1	2
2	contém	-	2	-	-	1	2
3	enormes	2	-	-	-	1	2
4	e	-	2	-	2	1	2
5	errantes	-	-	2	-	1	2
6	estrelas	-	-	-	-	4	0
7	fixas	-	-	2	-	1	2
8	galáxias	-	2	-	-	1	2
9	planetas	-	0,415	0,415	1,073	3	0,415
10	são	2	-	2	-	2	1
11	sóis	1	1	-	-	2	1
12	tem	1	-	-	2	2	1
13	órbitas	-	-	-	2	1	2
	Norma $\|\vec{d_i}\|$	3,162	3,629	3,489	4,141		

Fonte: Medeiros, 2021.

Utilizando-se a consulta para os termos “planetas errantes”, o cálculo pode ser visto no Quadro 1. A soma no numerador é referente ao produto interno ($tf - idf_j$ vezes idf_q) para cada termo da busca. O fator referente à norma do termo de busca (o segundo fator, a raiz do denominador da equação (3)) será $|\vec{q}| = \sqrt{2^2 + 0{,}415^2} = 2{,}042$.

Quadro 1 - Documentos ranqueados pelo modelo vetorial para a consulta dos termos “planetas errantes” utilizando os pesos TF-IDF

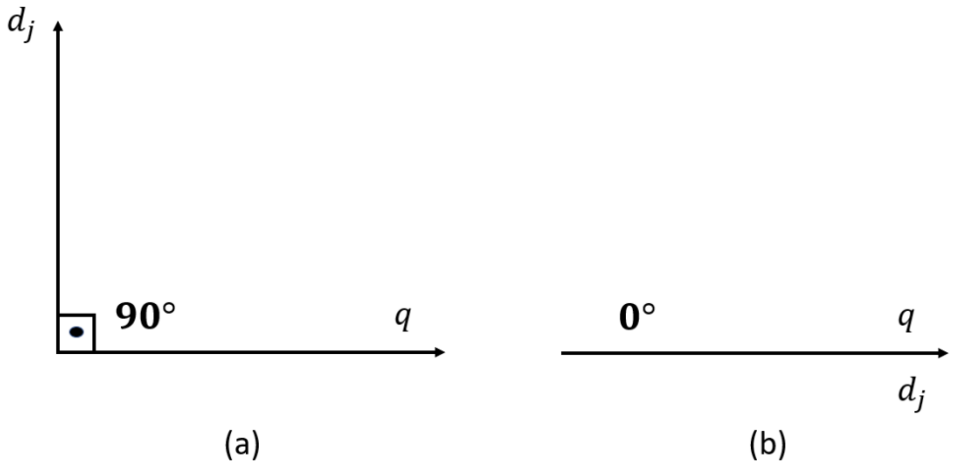
Documento	Cálculo	Rank
d_1	$\frac{0\,.\,2\,+\,0{,}415\,.\,0}{3{,}162\,.\,2{,}042} = \frac{0}{6{,}459}$	0,000
d_2	$\frac{0\,.\,2\,+\,0{,}415\,.\,0{,}415}{3{,}629\,.\,2{,}042} = \frac{0{,}172}{7{,}413}$	0,023
d_3	$\frac{2\,.\,2\,+\,0{,}415\,.\,0{,}415}{3{,}489\,.\,2{,}042} = \frac{4{,}172}{7{,}126}$	0,585
d_4	$\frac{0\,.\,2\,+\,0{,}415\,.\,1{,}073}{3{,}162\,.\,2{,}042} = \frac{0{,}445}{8{,}459}$	0,053

Fonte: Medeiros, 2021.

Pode-se notar, portanto, que os valores do rank em ordem decrescente ficam $\{d_3, d_4, d_2, d_1\}$, sendo o documento d_3 que contém os dois termos apresentado em primeiro lugar.

Os valores do rank estão normalizados, ou seja, estão dentro da faixa $\{0{,}1\}$. No modelo vetorial, o rank é entendido como o valor do cosseno entre o ângulo formado pelo vetor d_j pelo termo de busca q. Desta forma, quanto mais próximo de zero o valor do rank, mais ortogonal é o ângulo entre eles, ou seja, $90°$. Quanto mais próximo de 1, as retas tendem a ser coincidentes, próximas do ângulo $0°$ (Ver Figura 2).

Figura 2 - No modelo vetorial, (a) o ângulo de $90°$ indica nenhuma correlação, enquanto o ângulo $0°$ (b) indica alta correlação entre os termos de indexação com os termos de busca



Fonte: Medeiros, 2021.

1.2 Modelo Probabilístico

Como visto anteriormente, o modelo probabilístico é um modelo que considera o processo de recuperação de documentos considerando uma descrição de probabilidades de se encontrar os termos de busca nos documentos de uma coleção. Portanto, o modelo tenta estimar a probabilidade de que o usuário irá encontrar um documento que seja de interesse. A partir de uma consulta, existe um conjunto de documentos que contém exatamente os documentos relevantes e nenhum outro. O cálculo do grau de similaridade se dá mediante a razão dos documentos que são relevantes para o usuário em relação aos documentos que não são.

Considere R como o conjunto de documentos conhecidos ou pressupostos como relevantes para o usuário em uma consulta q, e $\bar{R}$ consistindo no complemento de R, ou seja, o conjunto de documentos não relevantes. Então $P(R|\overrightarrow{d_j}, q)$ é a probabilidade de que o documento d_j com representação $\overrightarrow{d_j}$ seja relevante para a consulta q. Por outro lado, $P(\bar{R}|\overrightarrow{d_j}, q)$ é a probabilidade de que o documento d_j não seja relevante para a consulta q. O grau de similaridade $sim(d_j, q)$ do documento d_j para a consulta q pode então ser calculado:

$$sim(d_j, q) = \frac{P(R|\overrightarrow{d_j}, q)}{P(\bar{R}|\overrightarrow{d_j}, q)} \tag{4}$$

Da mesma forma que a aplicação para o modelo vetorial, o modelo probabilístico será obtido mediante o uso da fórmula alternativa:

$$sim(d_j, q) \cong \sum_{k_i \in q \wedge k_i \in d_j} \log\left(\frac{N + 0.5}{n_i + 0.5}\right) \tag{5}$$

O Quadro 2 mostra o desenvolvimento do cálculo para o modelo probabilístico. Para o termo “planetas”, $n_9 = 3$ e para o termo “errantes”, $n_5 = 1$ (O índice está de acordo com a Figura 1). Para o documento d_1, os dois termos não aparecem; portanto, o rank é zero. Para os documentos d_2 e d_4, encontra-se o termo “planetas”, sendo que os dois termos ocorrem concomitantemente no documento d_3.

Quadro 2 - Documentos ranqueados pelo modelo probabilístico descrito no texto para a consulta dos termos “planetas errantes”

Documento	Cálculo	Rank
d_1	-	0,000
d_2	$\log_2 \frac{4+0,5}{3+0,5}$	0,362
d_3	$\log_2 \frac{4+0,5}{3+0,5} + \log_2 \frac{4+0,5}{1+0,5}$	1,947
d_4	$\log_2 \frac{4+0,5}{3+0,5}$	0,362

Fonte: Medeiros, 2021.

Os documentos em ordem decrescente seriam, portanto, $\{d_3, d_2, d_4, d_1\}$ ou $\{d_3, d_4, d_2, d_1\}$. Caso algum termo venha a ocorrer em todos os documentos, o cálculo do logaritmo para ele será também zero. Por exemplo, caso o termo de busca considerasse “estrelas”, tal termo aparece em todos os documentos; desta forma, $n_6 = 4$ e o cálculo seria $\log_2 \frac{4+0,5}{4+0,5} = \log_2 1 = 0$.

TEMA 2 – MÉTRICAS PARA COMPARAÇÃO DE PALAVRAS – DISTÂNCIA DE LEVENSHTEIN

No estudo das gramáticas constatou-se que o léxico (símbolos terminais) armazena as palavras e termos importantes para uma linguagem. A grafia dos símbolos é feita no momento da elaboração da gramática, não se modificando durante o uso. Quando tais palavras forem encontradas em um texto a ser analisado, por exemplo, o processamento de uma conversação em um *chatbot*, elas serão comparadas diretamente e, se houver o *match* (“casamento”) com a sua grafia, serão categorizadas na sua classe respectiva.

O problema acontece quando a grafia de uma palavra é feita de maneira equivocada ou errônea. Por exemplo, as palavras “casa” e “caça” são palavras com sentidos diferentes e se forem encontradas em um texto em que estejam trocadas, deixarão a frase sem sentido. Em outro caso, as palavras “casa” e “ccasa” tem a intenção do mesmo sentido, porém com grafia errada no segundo caso. Tais erros são comuns de acontecer em situações de conversação e trazem dificuldades adicionais ao processamento de linguagem natural.

Uma alternativa para o tratamento destas situações é considerar o *match* aproximado de palavras, ao invés do *match* determinístico. Isto pode ser conseguido mediante a aplicação de métricas que avaliem o quão próximas estão uma palavra ou frase de outra semelhante, ou que deveria ser igual. De forma geral, as métricas buscam fazer uma avaliação de características como a proximidade das letras, bem como a ordem em que estão colocadas de forma relativa.

Portanto, uma métrica permite o cálculo da **distância** entre termos. Com base na geometria, em que o cálculo da distância entre dois pontos pode ser feito utilizando-se a distância euclidiana, por exemplo, em NLP calcula-se a distância que um termo ou frase tem de outro termo.

As **métricas de termos**[1] (*strings*) ou **métricas de similaridade** são a classe de métricas baseadas em texto que resultam em uma pontuação de similaridade ou dissimilaridade (ou seja, a distância) entre dois termos em um texto. Uma métrica de termos fornece um número de ponto flutuante que indica o nível de similaridade com base na correspondência lexicográfica simples (Chen, 2012, p. 4). Por exemplo, a semelhança entre “maçã” e “maça” pode ser considerada muito mais do que “maçã” e “laranja” usando métricas de termos.

As métricas de termos têm sido amplamente aplicadas em várias áreas, como **correspondência aproximada de termos**, **comparação** e **busca *fuzzy***. A **verificação ortográfica** tornou-se uma das aplicações mais comuns. A partir de um documento de entrada, um corretor ortográfico precisa encontrar palavras que foram digitadas incorretamente pesquisando em seu dicionário. Um corretor ortográfico recomenda possíveis candidatos correspondentes semelhantes às palavras que não estão no dicionário com base nas métricas de termos. A **limpeza de dados** é outra aplicação significativa. As informações de diferentes fontes de dados geralmente apresentam várias inconsistências. A mesma entidade do mundo real pode ser representada em formatos ligeiramente diferentes. A limpeza de dados mantém a consistência das informações de diferentes fontes de dados (Chen, 2012, p. 4).

As métricas de termos podem ser subdivididas em três classes (Chen, 2012, p. 5–11):

[1] Para fins de praticidade na tradução, a palavra “*string*” foi substituída por “termo”. Além de palavras simples, as métricas podem se referir a termos compostos também.

1) **Métricas de edição:** são uma das classes mais importantes que permitem a determinação da distância entre dois termos calculando o custo da melhor sequência de operações de edição para se converter um termo em outro. Estas operações podem ser inserção de caracteres, exclusão, substituição e transposição. Existem várias alternativas para calcular a distância por métricas de edição, por exemplo, a distância de Levenshtein, distância de Hamming, distância de Jaro, dentre outras.
2) **Métricas *Q-Gram*:** uma *q-gram* é um subtermo consecutivo de um tamanho específico que pode ser usado como assinatura para todo o termo. *Q-gram* é normalmente usada na correspondência aproximada de termos, deslizando uma janela de comprimento q sobre os caracteres de um termo para criar um número de subtermos. Como *q-gram* pode ter menos de *q* caracteres, os caracteres "#" e "%" são usados para estender o termo prefixando-o com $q - 1$ ocorrências de "#" e sufixando-a com $q - 1$ ocorrências de "%". Quando q é igual a 1, o *q-gram* coincide com a métrica de edição.
3) **Métricas vetoriais:** a similaridade do cosseno é uma medida de similaridade baseada em vetor (como estudado anteriormente em recuperação da informação). O cosseno de dois vetores a, b pode ser derivado usando a fórmula do produto escalar $a.b = |a||b| \cos\theta$. Os termos de entrada a serem comparados são transformados em vetores e o grau de similaridade é obtido como o cosseno do ângulo entre os vetores.

A seguir, serão estudadas algumas métricas de edição por meio de exemplos.

2.1 Distância de Levenshtein

A distância de Levenshtein, uma das mais utilizadas, é uma medida da semelhança entre dois termos, o termo de origem s e o termo de destino t. A distância é o número de inserções, exclusões ou substituições de caracteres necessárias para transformar s em t. Quanto maior a distância de Levenshtein, mais diferentes são os termos entre si (Haldar; Mukhopadhyay, 2011).

O algoritmo para o cálculo da distância de Levenshtein é mostrado na Figura 3. De forma geral, na inicialização é identificado o tamanho dos termos

de entrada e a construção da matriz com as dimensões dos termos. No processamento, dentro do loop duplo, caso o caractere da coluna seja o mesmo da linha e assuma a variável, o custo será zero; se for diferente, será 1. Para preencher o elemento $d[i,j]$, este será a escolha do valor mínimo entre o valor acima, o valor à esquerda e o valor na diagonal (acima e à esquerda) mais o custo calculado anteriormente. O valor da distância de Levenshtein será o valor da úlgima célula calculado, ou seja, $d[n,m]$.

Figura 3 - Algoritmo para cálculo da distância de Levenshtein

Entrada: termos s, t

Passo 1 – Inicialização
a) Variáveis
n = comprimento de s
m = comprimento de t

b) Matriz
Construir matriz $d[0..n, 0..m]$
Primeira linha da matriz = $d[0, 0..n] = 0..n$
Primeira coluna da matriz = $d[0..m, 0] = 0..m$

Passo 2 - Processamento
Para $s[i],\ i = 1..n$
    Para $t[j], j = 1..m$
        Se $s[i] == t[j]$:
            $custo = 0$
        Senão
            $custo = 1$
        Faça $d[i,j]$ = mínimo entre:
            $(d[i-1,j] + 1,$ // célula acima mais um
            $d[i,j-1] + 1,$ // célula à esquerda mais um
            $d[i-1,j-1] + custo)$ // célula acima e à esquerda
            // (diagonal) mais o custo

Passo 3 - Resultado
Retorne $d[n,m]$

Fonte: Medeiros, 2021 baseado em Haldar; Mukhopadhyay, 2011.

Mostrando o funcionamento do algoritmo por meio de um exemplo, imagine calcular a distância de Levenshtein entre as palavras “CAMPO” e “CANTOR”. Como se pode notar, para transformar a primeira palavra na segunda, será necessário fazer duas substituições e uma inserção. Na Figura 4 consta o exemplo inicializado, mostrando o preenchimento da primeira linha e da primeira coluna.

Figura 4 - Inicialização da matriz para o exemplo do texto

		C	A	M	P	O
	0	1	2	3	4	5
C	1					
A	2					
N	3					
T	4					
O	5					
R	6					

Fonte: Medeiros, 2021.

Em seguida, executa-se para cada linha e coluna o preenchimento dos valores da matriz. O próximo valor será zero pela coincidência das letras “C”. Na célula da mesma coluna e abaixo, o valor será o mínimo escolhido entre 1+1=2 (acima), 2+1=3 (esquerda) e 0+1=1 (diagonal), sendo, portanto, 1. E assim, sucessivamente, como mostrado na Figura 5.

Figura 5 - Execução da segunda coluna para o exemplo do texto

		C	A	M	P	O
	0	1	2	3	4	5
C	1	0				
A	2	1				
N	3	2				
T	4	3				
O	5	4				
R	6	5				

Fonte: Medeiros, 2021.

Na coluna referente à letra “A” da palavra “CAMPO” e a letra “C” da palavra “CANTOR”, o valor a ser escolhido será o mínimo entre 2+1=3 (acima), 0+1=1 (esquerda) e 1+1=2 (diagonal), sendo, portanto, 1. A célula logo abaixo será zero, pela coincidência das letras “A”. Na célula indexada pela letra “N” da palavra “CANTOR”, deverá ser colocado o valor mínimo entre 0+1=1 (acima), 2+1=3 (esquerda) e 1+1=2 (diagonal), ou seja, 1. E assim sucessivamente. A execução desta coluna é vista na Figura 6.

Figura 6 - Execução da terceira coluna para o exemplo do texto

		C	A	M	P	O
	0	1	2	3	4	5
C	1	0	1			
A	2	1	0			
N	3	2	1			
T	4	3	2			
O	5	4	3			
R	6	5	4			

Fonte: Medeiros, 2021.

Da mesma forma, continua-se o processo para as colunas seguintes, chegando-se ao último valor indexado pela letra “O” da palavra “CAMPO” e pela letra “R” da palavra “CANTOR”. Este valor será a distância de Levenshtein. Na Figura 7, com o cálculo concluído, o valor 3 pode ser identificado. Em alguns casos pode ser calculado um **grau de similaridade**, dividindo-se o valor da distância encontrado pelo máximo tamanho da linha ou coluna, e diminuindo de 1. No caso o maior termo tem tamanho 6; assim, o valor normalizado seria $1 - 3/6 = 0{,}5$. Quanto mais próximo um termo estiver de outro, mais próximo de 1 o grau de similaridade pela distância de Levenshtein será; quanto mais diferentes forem os termos comparados, mais próximo de zero será o grau de similaridade.

Figura 7 - Término do cálculo da distância de Levenshtein, resultando no valor 3

		C	A	M	P	O
	0	1	2	3	4	5
C	1	0	1	2	3	4
A	2	1	0	1	2	3
N	3	2	1	1	2	3
T	4	3	2	2	2	3
O	5	4	3	3	3	2
R	6	5	4	4	4	3

Fonte: Medeiros, 2021.

Uma variante da distância de Levenshtein é calculada pelo **algoritmo de Damerau-Levenshtein**, o qual permite também mais uma operação de edição: a transposição de caracteres adjacentes (Chen, 2012, p. 8).

TEMA 3 - MÉTRICAS DE JARO E JARO-WINKLER

A métrica de Jaro mede a distância de edição entre duas sequências ou termos. A métrica Jaro-Winkler é uma variante da métrica Jaro e é amplamente utilizada nas áreas de extração de informações, vinculação de registros, vinculação de entidades, uma vez que funciona bem na correspondência de nomes pessoais e de entidades. A métrica[2] permite calcular um grau de similaridade entre duas sequências. Quanto maior o grau de similaridade de Jaro ou Jaro-Winkler entre duas sequências ou termos, maior a probabilidade de eles serem semelhantes. O limite de pontuação é 0 para não semelhante e 1 para semelhante (Cahyono, 2019).

Este comparador de termos foi introduzido pela primeira vez por Jaro (1989). Considerando as operações de inserção, exclusão e transposição. Os principais componentes do algoritmo de distância de Jaro são:

- Cálculo dos comprimentos dos termos.
- Identificação de caracteres comuns nos dois termos.
- Contagem de transposições de caracteres.

Conforme Winkler (2006), a definição de caractere comum é que o caractere concordante deve estar dentro da metade do comprimento da sequência mais curta. A definição de transposição é que o caractere d[illegible] (6)
sequência está fora de ordem com o caractere comum correspondente d
sequência. O grau de similaridade de Jaro, já normalizado é:

$$sim_J(s_1, s_2) = \frac{1}{3}\left(\frac{n_c}{l_1} + \frac{n_c}{l_2} + \frac{0.5n_t}{n_c}\right)$$

Na fórmula (6), s_1 e s_2 são as sequências com comprimentos l_1 e l_2, n_t é o número de transposições e n_c é a quantidade de caracteres comuns entre as duas sequências, em que a distância para caracteres comuns está dentro da metade do comprimento mínimo de s_1 e s_2: $\left\lfloor \frac{\max(l_1,l_2)}{2} \right\rfloor - 1$. Existem variações da fórmula (6) na literatura, dando-se preferência aqui para a que foi utilizada diretamente por Winkler.

[2] No sentido estrito, a métrica de Jaro não é propriamente uma métrica conforme a geometria euclidiana, pois não segue a desigualdade triangular. Por isso, alguns autores mencionam “similaridade” ao invés de “distância” ou “métrica”.

Exemplo: calcular o grau de similaridade de Jaro para os termos "PRATO" E "TRAPO".

Fazendo-se $s_1 = PRATO$ e $s_2 = TRAPO$, pode-se ver que as duas palavras têm comprimento $l_1 = l_2 = 5$. Todos os caracteres são comuns. Entretanto, como é necessário considerar aqueles que são comuns dentro da distância $\left\lfloor \frac{\max(l_1,l_2)}{2} \right\rfloor - 1 = \left\lfloor \frac{\max(5,5)}{2} \right\rfloor - 1 = \left\lfloor \frac{5}{2} \right\rfloor - 1 = \lfloor 2{,}5 \rfloor - 1 = 2 - 1 = 1.$ Ou seja, os caracteres comuns não podem estar a mais do que 1 caractere de distância; portanto, considera-se comuns apenas R, A e O e $n_c = 3$. As transposições são $P \to T$ e $T \to P$, ou seja, $n_t = 2$. Agora, tem-se todos os elementos para calcular a distância de Jaro:

$$sim_J(s_1, s_2) = \frac{1}{3}\left(\frac{n_c}{l_1} + \frac{n_c}{l_2} + \frac{0.5n_t}{n_c}\right) = \frac{1}{3}\left(\frac{3}{5} + \frac{3}{5} + \frac{0{,}5\,.\,2}{3}\right) = \frac{23}{45} = 0{,}511.$$

O cálculo da similaridade de Jaro-Winkler, usa a similaridade Jaro combinada com um fator constante de escala p, a qual permite obter classificações mais favoráveis para sequências que casam desde o início, para um comprimento de prefixo definido ℓ:

$$sim_{JW} = sim_J + \ell p(1 - sim_J) \tag{7}$$

O fator de escala p pode ser entendido como um percentual de incremento que a similaridade terá devido ao casamento das partes iniciais das sequências (p não deve exceder $0{,}25$; Winkler sugere utilizar $0{,}1$). O comprimento ℓ não deve ser maior do que 4. Para o exemplo anterior, adotando eventualmente uma escala $p = 0{,}1$ para um comprimento $\ell = 3$, obtém-se:

$$sim_{JW} = 0{,}511 + 3\,.\,0{,}1\,.\,(1 - 0{,}511) = 0{,}664.$$

Resumindo, o cálculo da distância de Levenshtein e das métricas de Jaro e Jaro-Winkler são encontrados amplamente na literatura sobre métricas de comparação de termos. Outras métricas podem ser encontradas, por exemplo, a distância de Hamming (Chen, 2012), que pode ser utilizada apenas com sequências de tamanhos iguais; ou a distância de Jaccard (Chung et al., 2019), a qual usa a teoria de conjuntos para o cálculo da similaridade.

TEMA 4 – TRADUÇÃO AUTOMÁTICA

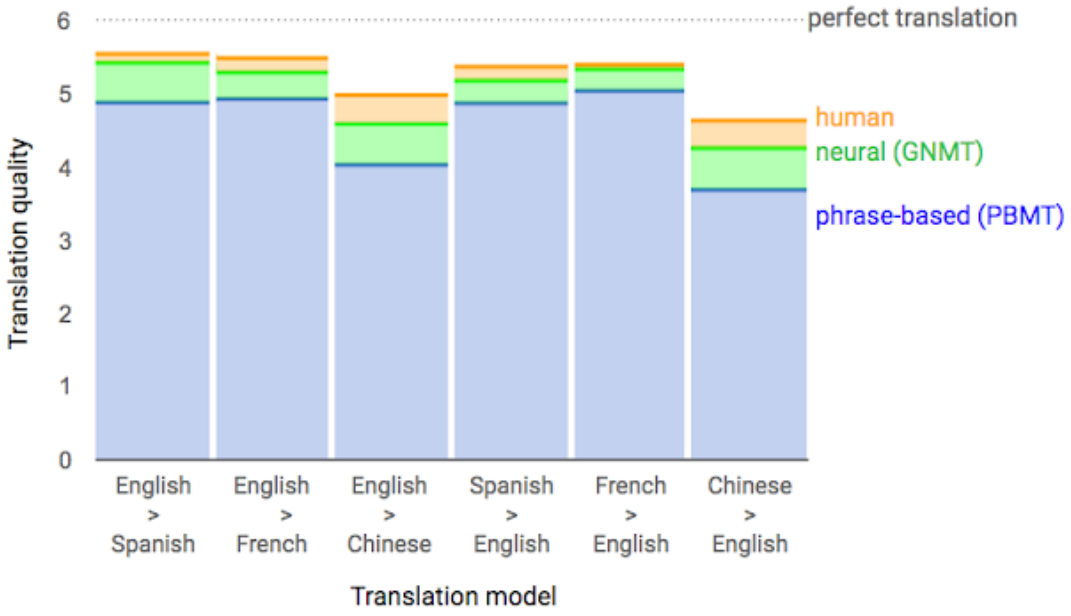
Ainda na década de 1960 existia uma grande expectativa de que sistemas computacionais fossem capazes de fazer a tradução de textos de uma linguagem natural para outra, induzido de certa maneira pelo processo de decriptação da máquina Enigma, no projeto de Turing, para decifrar as mensagens alemãs na Segunda Guerra Mundial. Entretanto, a tradução se mostrou uma tarefa difícil, pois além da correspondência entre palavras de linguagens diferentes, há a questão semântica envolvida na tradução, ou seja, a exigência de uma compreensão profunda do texto a ser traduzido (Russell; Norvig, 2004, p. 823).

Existem três abordagens principais para resolver o problema de gerar tradução automática (Araújo; Pereira; Benevenuto, 2020, p. 1085):

- **Baseado em regras/baseado em frases:** utiliza léxicos combinados com definições gramaticais para traduzir sentenças de forma significativa.
- **Métodos estatísticos:** tenta construir um modelo de tradução analisando uma grande quantidade de dados de treinamento para cada par de idiomas.
- **Redes neurais artificiais:** constroem uma grande rede neural artificial usando uma enorme quantidade de dados de treinamento, essa abordagem tornou-se popular recentemente e mostra melhor desempenho de tradução.

Na Figura 8 é mostrado um gráfico de desempenho, comparando três candidatos a tradutores: uma rede neural, um sistema baseado em frases e humanos adequados. O gráfico demonstra o quão próximos os atuais sistemas de tradução automática de última geração estão dos tradutores humanos. Além disso, mostra que as redes neurais parecem superar a estratégia baseada em frases (Araújo; Pereira; Benevenuto, 2020).

Figura 8 - Comparação entre as técnicas baseadas em rede neural artificial e as baseadas em frase, com uma linha de base humana e a linha da tradução considerada perfeita



Fonte: extraída de Le; Schuster, 2016.

No Quadro 3 estão mostrados alguns sistemas de tradução automática.

Quadro 3 - Exemplos de sistemas de tradução automática

Sistema	Descrição
Google Translator	A versão anterior do Google Translator costumava ser baseada em frases e usa o inglês como idioma intermediário para a tradução. No entanto, agora ele utiliza Redes Neurais e tradução direta em pares de idiomas, segundo os autores, essa nova abordagem é responsável por melhorar o desempenho do sistema em 55% em comparação com a versão baseada em frases.
Microsoft Translator Text API	Desde 2010, a Microsoft usa redes neurais em seus sistemas de tradução. Dado qualquer par de idiomas para traduzir, o sistema utiliza características únicas do par que apresenta um vetor de 500 dimensões. Codifica conceitos como gênero (feminino, masculino, neutro),

	nível de polidez (gíria, casual, escrita, formal etc.), tipo de palavra (verbo, substantivo etc.) e outras características não óbvias.
Yandex Translate API	A tradução automática Yandex é baseada na abordagem estatística. Para aprender um idioma, o sistema compara centenas de milhares de textos paralelos que se traduzem “frase por frase”. Ele tem dois componentes principais: o modelo de tradução e o modelo baseado em regras.

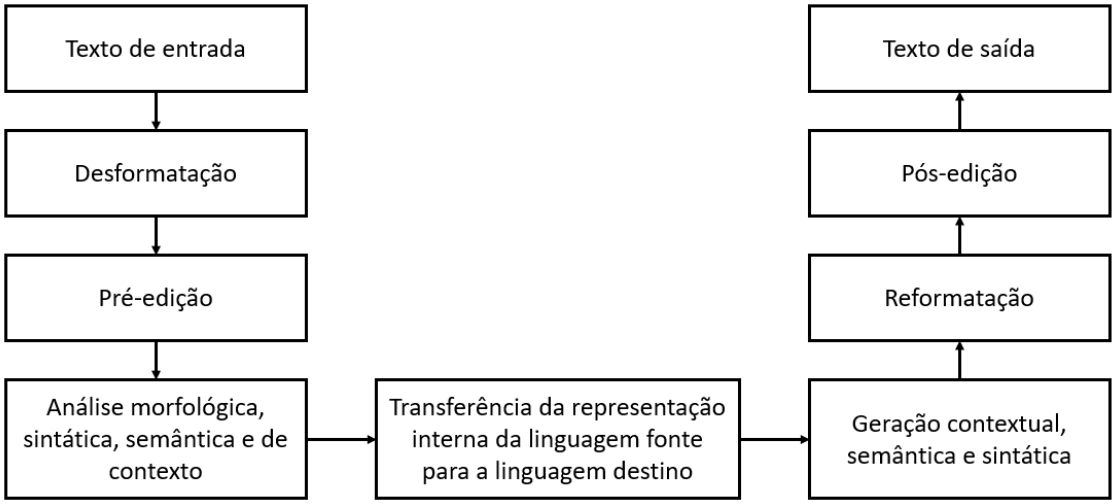
Fonte: elaborado com base em Araújo et al., 2020.

Hoje em dia, a maioria das tecnologias de tradução automática utiliza uma abordagem híbrida, aproveitando os métodos e técnicas anteriores aplicados até o momento. Vale ressaltar que, embora tenham surgido na década de 1950, os métodos de tradução automática baseados em regras apresentam bom desempenho entre idiomas muito semelhantes que poderiam ser considerados dialetos um do outro. Desta forma, pode-se afirmar que a tradução automática evoluiu a partir de diferentes abordagens, e as redes neurais, a última geração de sistemas de TA, está rapidamente se tornando a abordagem dominante, direcionada para dados (Esqueda, 2021, p. 283).

No entanto, dados de treinamento suficientes para um sistema com rede neural estão disponíveis apenas para um punhado, talvez vinte, das mais de quatro mil línguas do mundo. Para o resto (mais de 99%), ou abordagens baseadas em regras ou de tradução automática estatística, ou, na maioria dos casos, a tradução humana, são as únicas opções (Melby, 2020).

Na Figuras 9 e no Quadro 4 está ilustrado e descrito um diagrama contendo um processo padrão de tradução de máquina. Ainda que a ideia de um tradutor contemple a existência de um dicionário que permite a representação das palavras de uma linguagem em outra, o processo deve contemplar um pré-processamento que envolva tanto os aspectos de extrair os textos de formatos específicos ou impróprios, bem como aqueles relacionados às análises de linguagem que devem fazer parte: morfológica, sintática, semântica e de contexto.

Figura 9 - Fluxograma do processo de tradução automática



Fonte: elaborado com base em Albadr; Tiun; Al-Dhief, 2018.

Quadro 4 - Descrição do processo padrão de tradução de máquina

Fase	Descrição
Texto de entrada	As frases são classificadas de acordo com seu nível de complexidade em termos de tradução. O desempenho dos sistemas MT é significativamente prejudicado por frases que vêm com associações, antecipações, suposições e estipulações. Certas frases podem exigir o envolvimento tanto do conhecimento do mundo quanto do senso comum para a realização de uma interpretação apropriada.
Desformatação e reformatação	Servem para suavizar e elevar a qualidade do procedimento de tradução automática. Os segmentos no idioma de origem que requerem tradução precisam ser distinguidos. Entradas como números e fluxogramas não exigem tradução e podem ser excluídos.
Pré-edição/ Pós-edição	A eficácia de um sistema de TA é determinada por sua qualidade de pré-edição e pós-edição. Vários sistemas de TM exigem a condensação de frases longas, seccionando-as em frases mais curtas. A pré-edição também implica resolver a questão dos sinais de pontuação e a exclusão de itens que não requerem tradução. O objetivo da publicação é garantir que a qualidade da tradução seja de um padrão aceitável. A pós-edição é particularmente importante durante a tradução de informações vitais, como aquelas relacionadas a questões de saúde.
Análise, transferência e geração	A forma da palavra de inflexões, tempos, números, partes do discurso etc. são determinados por meio de uma análise morfológica. A análise sintática verifica se as palavras em uma frase são de natureza subjetiva ou objetiva. O papel da análise semântica e contextual, por outro lado, é aproveitar os resultados da análise sintática para a construção de uma frase traduzida apropriadamente. Análises sintáticas e semânticas são frequentemente realizadas em uníssono para gerar correspondentemente uma configuração de árvore sintática e uma rede semântica. Esse processo leva à formação da

	estrutura interna da frase. Para a fase de geração de sentenças, o processo de análise é simplesmente invertido.
Análise morfológica e geração	A morfologia computacional lida com a identificação, exame e geração de palavras. Os procedimentos morfológicos abrangem flexão, derivação, afixação e composição. Dentre esses procedimentos, a flexão não é apenas o mais empregado, mas também o mais prolífico. A tarefa atribuída à flexão é a revisão da forma da palavra para número, gênero, modo, tempo, aspecto, pessoa e caso. Um analisador morfológico serve para fornecer detalhes sobre as palavras que examina.
Análise sintática e geração	Enquanto as palavras são consideradas o núcleo para procedimentos relacionados ao processamento de fala e linguagem, a sintaxe é considerada a estrutura. A análise sintática tem a ver com a maneira pela qual (a) as palavras são separadas em categorias conhecidas como partes do discurso, (b) as palavras se juntam a palavras vizinhas para formar frases e (c) as palavras em uma sentença dependem de cada de outros.
Análise semântica, análise contextual e geração	A análise semântica constrói as representações de significado e atribui-lhes as entradas linguísticas. Os analisadores semânticos utilizam o léxico e a gramática para a geração de significados que não são influenciados pela relevância contextual. A fonte de informação compreende o significado das palavras, significados relacionados às configurações gramaticais, detalhes sobre o contexto do discurso e conhecimento do senso comum.

Fonte: elaborado com base em Albadr; Tiun; Al-Dhief, 2018.

TEMA 5 – ANÁLISE DE SENTIMENTO E MINERAÇÃO DE OPINIÃO

A análise de sentimentos reúne várias áreas de pesquisa, como processamento de linguagem natural, mineração de dados e mineração de texto, e está rapidamente se tornando de grande importância para as organizações que se esforçam para integrar métodos de inteligência computacional em suas operações e tentam esclarecer e melhorar seus produtos e serviços. Na análise de sentimento, ou mineração de opinião (*Sentimental Analysis or Opinion Mining* - SAOM), o objetivo é descobrir as opiniões das pessoas expressas em linguagem escrita (texto). Sentimento no termo significa “o que se sente sobre algo”, “experiência pessoal, o próprio sentimento”, “uma atitude em relação a algo” ou “uma opinião” (Farhadloo; Rolland, 2016).

A seguir, alguns termos são importantes dentro do contexto de análise de sentimento para a compreensão dos objetivos e das tarefas que estão envolvidas neste tema (Farhadloo; Rolland, 2016):

- **Colaborador:** um colaborador é a pessoa ou organização que está expressando suas opiniões em linguagem escrita ou texto. O colaborador também pode ser chamado de detentor de opinião ou fonte de opinião.
- **Objeto:** um objeto é uma entidade que pode ser um produto, serviço, pessoa, evento, organização ou tópico. Pode estar associado a um conjunto de componentes e atributos que na literatura de análise de sentimentos esses componentes e atributos são chamados de aspectos.
- **Revisão:** uma revisão é um texto gerado pelo colaborador que contém as opiniões do colaborador sobre alguns aspectos do objeto. Uma revisão também pode ser chamada de documento opinativo.
- **Classificação geral:** esta é uma satisfação geral relatada pelo usuário com o objeto, por exemplo, em uma escala Likert de 1 a 5.
- **Opinião:** uma opinião sobre um aspecto é uma visão, atitude, emoção ou avaliação positiva, neutra ou negativa de um colaborador.
- **Aspecto:** um aspecto é um atributo importante do objeto em relação à satisfação geral do cliente que o colaborador comentou em sua revisão.

Uma **revisão opinativa** pode ser gerada por vários colaboradores ou conter opiniões de várias fontes. Também deve-se notar que a revisão pode ser uma revisão direta sobre um único objeto ou uma revisão comparativa em que compara dois ou mais objetos entre si. Em geral, uma revisão d_i é uma coleção de sentenças $d_i = \{s_{i1}, s_{i2}, \ldots, s_{im}\}$ que contém opiniões de diferentes contribuidores sobre diferentes aspectos de diferentes objetos.

Vale ressaltar que a maior parte da literatura tem considerado a orientação da opinião como uma variável **binária**: positiva ou negativa. Existem alguns trabalhos que incluem a classe neutra também. Em geral, a orientação de opinião pode ser considerada em diferentes escalas e não apenas binária ou ternária.

Portanto, o objetivo da análise de sentimentos ou mineração de opinião pode ser descrito como: *dada uma coleção de revisões* $D = \{d_1, d_2, \ldots, d_n\}$ *versando sobre um objeto, deve ser descoberto todos os aspectos e sentimentos correspondentes expressos nesta coleção* (Farhadloo; Rolland, 2016).

Para atingir este objetivo, pode-se considerar diferentes níveis:

- **Análise de sentimento em nível de documento:** caso o foco esteja em cada documento e a orientação de sentimento de todo o documento possa ser encontrada. Embora um documento possa transmitir um sentimento

geral positivo ou negativo, é bem provável que nem todas as frases de um documento sejam todas positivas ou negativas. Análises neste nível podem ser feitas em e-mails recebidos por uma empresa para sondar as percepções de clientes com relação a algum produto específico. Por exemplo, se as resenhas de interesse forem resenhas de filmes, um colaborador específico pode dar uma classificação geral positiva ao filme, no entanto, eles podem não gostar de todos os aspectos desse filme e, em sua resenha, mencionar os aspectos de que gostam e o que não gostam nesse filme.

- **Análise de sentimento em nível de sentença:** quando se encontra o foco da análise de sentimento em sentenças e uma orientação de sentimento para sentenças da revisão. As análises deste tipo são comuns com redes sociais que permitem expressões curtas, como o Twitter. No Quadro 5 está um exemplo de sentenças expressas em um site sobre um determinado produto (caixa organizadora), mostrando a orientação do sentimento e certos problemas relacionados com a interpretação. Quanto às afirmações 1, 2, 5 e 6, o sentimento expresso na sentença é bem específico e o analisador de sentimento pode identificar diretamente palavras do léxico como “bom”, “útil”, “gostar”. Pode ainda analisar expressões com negação para classificar como sentimento negativo. No entanto, expressões como 3, 4 e 7 são mais complexas de interpretar.

Quadro 5 - Exemplo de sentenças curtas sobre um produto retirado de um site

Nº	Sentença	Sentimento
1	“O produto é bom, bastante útil”	+
2	“A caixa veio com defeito na tampa, não fecha direito”	–
3	“Grande coisa”	?
4	“Em tamanho maior, dá para guardar coisas grandes”	?
5	“Deveria vir com etiqueta na frente”	–
6	“Gosto dos produtos desta empresa”	+
7	“Eu gostei do produto, mas minha esposa não”	?

Fonte: Medeiros, 2021.

- **Análise de sentimento em nível de aspecto:** em uma análise mais apurada, pode-se estar interessado em saber quais aspectos particulares

do objeto o colaborador está comentando e também se ele gosta deles ou não. A análise de sentimento neste nível demanda a extração de objetos dentro dos documentos ou sentenças, que pode ser feita de forma automática (Farhadloo; Rolland, 2016).

Alguns desafios devem ser levados em consideração em se tratando de análise de sentimentos. Os aspectos e problemas relacionados com as características da linguagem costumam impor restrições a abordagens generalistas. São descritos a seguir alguns tópicos (Farhadloo; Rolland, 2016):

- **Sinonímia e polissemia:** a questão da **sinonímia** se relaciona a usuários em diferentes contextos ou com diferentes necessidades, conhecimentos ou hábitos linguísticos descrevendo a mesma informação usando termos diferentes. Como as pessoas em sites de busca e autores costumam usar palavras diferentes, materiais relevantes podem ser perdidos. Já com relação à **polissemia**, as pessoas podem usar o mesmo termo para se referir a coisas diferentes. Em geral, a sinonímia e a polissemia são causadas devido à possível variabilidade no uso das palavras. As questões de sinonímia e polissemia apresentam desafios difíceis na análise de sentimentos e no problema de mineração de opinião. Para minimizar tais problemas, o ***stemming***, que visa extrair o radical das palavras, é uma técnica utilizada. O objetivo é reunir as variações de formas de uma palavra às suas raízes morfológicas. O uso de vocabulário controlado é outra alternativa para evitar a variabilidade. A **análise semântica latente** (LSA), que pode utilizar a matriz termo-documento, é outra abordagem para tratar da sinonímia.
- **Identificação de sarcasmo:** é uma tarefa muito difícil para os humanos e ainda mais difícil para as máquinas. A capacidade de identificar de forma confiável o sarcasmo no texto pode melhorar o desempenho de muitas tarefas de processamento de linguagem natural, particularmente para a análise de sentimento. O sarcasmo é uma forma de expressão em que o significado literal é oposto ao pretendido. “O restaurante foi ótimo na medida em que fará com que todas as refeições futuras pareçam mais deliciosas”, é um exemplo de frase sarcástica. Embora tecnicamente não haja termo negativo na linguagem, pretende-se transmitir um sentimento

negativo. O exemplo 4 e 5 do Quadro 5 traz sentenças que parecem ter sarcasmo, o que dificulta a interpretação do sentimento.

- **Sentenças compostas:** uma sentença composta possui duas cláusulas ou sentenças independentes. Duas orações independentes podem ser unidas por uma conjunção coordenativa (como “e”, “ou”, “mas” e “para”) ou ainda ponto e vírgula. Lidar com sentenças compostas torna o problema da análise de sentimento difícil. Por exemplo, frases como “As crianças gostaram da praia, mas nós não” ou “Apesar de uma experiência agradável, não posso apoiar as muitas críticas de que era um ótimo restaurante” são desafiadoras para a análise de sentimentos. O exemplo 7 do Quadro 5 ilustra o problema de sentenças compostas.
- **Dados não estruturados:** um desafio para a análise de sentimentos é a transformação dos dados de entrada não estruturados que estão disponíveis na forma de revisões escritas em dados semiestruturados. Dados semiestruturados são dados que não são dados brutos, nem estão em conformidade com a estrutura formal de modelos de dados associados a bancos de dados relacionais ou outras formas de tabelas de dados, mas ainda assim contêm *tags* ou outros marcadores que permitem separar elementos semânticos.
- **Identificação de aspectos:** o objetivo da identificação de aspecto é descobrir os aspectos particulares do objeto sobre os quais os colaboradores estão expressando suas opiniões. Existem duas categorias de trabalhos na literatura que abordam a identificação de aspectos: i) **extração automática**, quando não há conhecimento prévio sobre os aspectos e os aspectos são extraídos automaticamente das revisões fornecidas, podendo ser dividida em tarefas de extração supervisionadas e não supervisionadas; e ii) **extração semiautomática**, quando tanto um subconjunto de aspectos quanto todo o conjunto de aspectos desejados são conhecidos a priori. Nos casos em que um subconjunto de aspectos é conhecido, o subconjunto está sendo usado como um conjunto de primitivos, com possibilidades de expansão.
- **Identificação de sentimento:** a identificação do sentimento é o processo de descobrir a orientação da opinião dos fragmentos de texto de interesse. A orientação do sentimento pode ser expressa em diferentes escalas. As escalas de classificação são amplamente utilizadas on-line na tentativa

de fornecer indicações das opiniões dos consumidores sobre os produtos. O uso de escalas do tipo Likert 1 a 5 são bem comuns. Técnicas automáticas fazem uso de algoritmos de aprendizado de máquina para resolver o problema e esses algoritmos visam encontrar padrões de dados e tentar aprender com exemplos de treinamento. Quando não há exemplos de treinamento suficientes ou há exemplos de treinamento com variedade muito limitada, pode-se imaginar o quão difícil seria a extração de padrões. Para aprendizado supervisionado, algumas técnicas têm sido comuns como máquina de vetor de suporte (SVM) e classificação bayesiana (Naive-Bayes). No contexto de aprendizado profundo, o uso de LDA (*Latent Dirichlet Allocation*) para encontrar descrições curtas de elementos de uma descrição; e algoritmo baseados em amostragem como as cadeias de Markov – Monte Carlo (MCMC), têm sido utilizados.

FINALIZANDO

Esta aula teve o objetivo de mostrar a relação dos modelos de recuperação de informação com o esquema TF-IDF, particularmente mostrando o modelo vetorial e o probabilístico com base em exemplos. As métricas de comparação de palavras foram apresentadas como alternativa para a comparação aproximada de termos, sendo explicadas com exemplos a distância de Levenshtein, uma das mais utilizadas, e a métrica de similaridade de Jaro e Jaro-Winkler. A tradução automática, uma das tarefas da NLP, também foi apresentada mostrando o quanto este campo evoluiu ao longo dos anos e o quanto estamos próximos de sistemas de tradução quase perfeitos. Por fim, foi abordado a análise de sentimentos e mineração de opinião, mostrando uma área em crescimento dentro da NLP e as dificuldades relacionadas com a interpretação de textos que os usuários expressam e a sua classificação como opinião positiva ou negativa.

REFERÊNCIAS

ALBADR, M. A. A.; TIUN, S.; AL-DHIEF, F. T. Evaluation of machine translation systems and related procedures. **ARPN Journal of Engineering and Applied Sciences**, [s. l.], v. 13, n. 12, p. 3961–3972, 2018.

ARAÚJO, M.; PEREIRA, A.; BENEVENUTO, F. A comparative study of machine translation for multilingual sentence-level sentiment analysis. **Information Sciences**, [s. l.], v. 512, p. 1078–1102, 2020. Doi: <https://doi.org/10.1016/j.ins.2019.10.031>.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval**. 2. ed. Harlow, England: Addison Wesley, 2011.

CAHYONO, S. C. Comparison of document similarity measurements in scientific writing using Jaro-Winkler Distance method and Paragraph Vector method. **{IOP} Conference Series: Materials Science and Engineering**, [s. l.], v. 662, n. 5, p. 52016, 2019. Doi: <https://doi.org/10.1088/1757-899x/662/5/052016>.

CHEN, H. **String Metrics and Word Similarity Applied to Information Retrieval**. 2012. - University of Eastern Finland, [s. l.], 2012.

CHUNG, N. C. et al. Jaccard/Tanimoto similarity test and estimation methods for biological presence-absence data. **BMC bioinformatics**, [s. l.], v. 20, n. Suppl 15, p. 644, 2019. Doi: <https://doi.org/10.1186/s12859-019-3118-5>.

ESQUEDA, M. D. Machine Translation: Teaching and Learning Issues. **Trabalhos em Linguística Aplicada**, [s. l.], v. 60, n. 1, p. 282–299, 2021. Doi: <https://doi.org/10.1590/010318139320015202102l2>.

FARHADLOO, M.; ROLLAND, E. Fundamentals of sentiment analysis and its applications. **Studies in Computational Intelligence**, [s. l.], v. 639, n. August 2018, p. 1–24, 2016. Doi: <https://doi.org/10.1007/978-3-319-30319-2_1>.

HALDAR, R.; MUKHOPADHYAY, D. Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach. **CoRR**, [s. l.], v. abs/1101.1, 2011. Disponível em: <http://arxiv.org/abs/1101.1232>. Acesso em: 1 fev. 2022.

JARO, M. A. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. **Journal of the American Statistical Association**, [s. l.], v. 84, n. 406, p. 414–420, 1989. Disponível em: <http://www.jstor.org/stable/2289924>. Acesso em: 1 fev. 2022.

LE, Q. V.; SCHUSTER, M. **A neural network for machine translation, at production scale**. [S. l.], 2016. Disponível em: <https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html>. Acesso em: 18 jan. 2022.

MELBY, A. K. Future of Machine Translation: Musings on Weaver's memo. *In*: O'HAGAN, M. (org.). **The Routledge Handbook of Translation and Technology**. (1. ed. New York: Routledge, 2020. p. 419–436. *E-book*.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial - Tradução da 2. ed**. Rio de Janeiro: Editora Campus, 2004.

WINKLER, W. E. **Overview of Record Linkage and Current Research DirectionsBureau of the Census**. [S. l.: s. n.], 2006. Doi: <https://doi.org/10.1177/102538230701400102O1>.