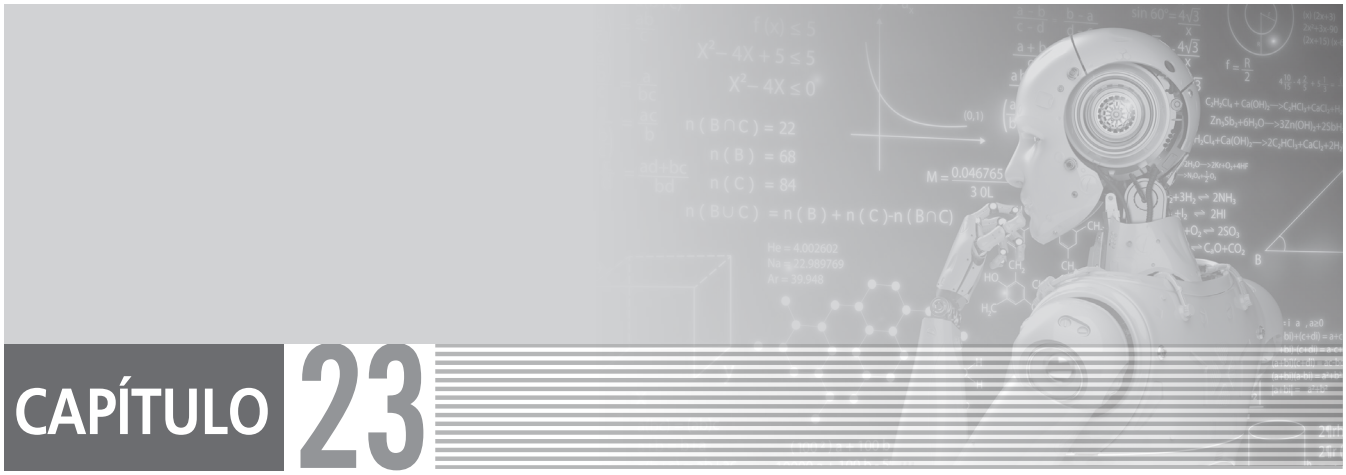

CAPÍTULO 23

CATEGORIZAÇÃO DE TEXTOS

A escrita ainda é uma das formas mais empregadas para registrar informações. Até pouco tempo, os textos eram escritos e armazenados somente em papéis, e o acesso e a recuperação das informações eram tarefas dispendiosas, pois dependiam exclusivamente da leitura e da organização humana. Com o avanço dos computadores, os textos agora podem ser armazenados em formatos digitais, o que trouxe grandes benefícios em relação ao espaço ocupado no armazenamento físico, facilidade no acesso, recuperação e disseminação da informação.

Em face das facilidades oferecidas pelo armazenamento e compartilhamento de textos em formato digital, um enorme volume de documentos é gerado e compartilhado a todo momento nos meios eletrônicos. Atualmente, livros, jornais e revistas, que até pouco tempo eram disponibilizados somente em papel, também contam com versões publicadas na internet, sendo que muitos nem são mais impressos. Nas empresas e até mesmo nas habitações, o uso de serviços informatizados para as tarefas do dia a dia, como os gerenciadores de *e-mail* e mensageiros instantâneos, fazem com que muitos textos estejam armazenados em arquivos digitais. Além disso, com a popularização do uso dos *smartphones*, muitas pessoas passaram a gerar conteúdo, como comentários e opiniões em *sites* e em redes sociais.

A crescente quantidade de documentos de texto produzidos trouxe novos desafios para a busca e a análise de seus conteúdos. Os seres humanos não são capazes de processar muitos documentos rapidamente por meio da leitura, o que torna cada vez mais necessário o emprego de técnicas automatizadas e eficientes para o processamento e a recuperação de informação. Uma maneira de auxiliar nessa tarefa é a partir da *categorização de textos*, também conhecida como classificação de textos. Trata-se da tarefa de definir categorias ou rótulos aos documentos textuais com a finalidade de definir documentos que compartilham características semelhantes. Dessa forma, é possível selecionar os documentos que possam conter informações relevantes verificando apenas a categoria na qual eles pertencem, sem precisar consultar e ler todos os documentos existentes.

A categorização de textos é amplamente empregada com sucesso em diversas aplicações do cotidiano, como análise de opinião (Bhowmick et al., 2010; Muhammad et al., 2016; Lochter et al., 2016), categorização de *e-mails* e detecção de *spam* (Almeida et al., 2011a,b; Almeida e Yamakami, 2012a,b,c; Alberto et al., 2015a,b; Almeida et al., 2016; Almeida e Yamakami, 2016; Silva et al., 2017b; Lochter et al., 2018b), classificação de notícias (Gutlein et al., 2009; Rossi et al., 2016; Aphinyanaphongs et al., 2014; Wu et al., 2016;

Jiang et al., 2016), detecção de opiniões e notícias falsas (Cardoso et al., 2018; Monteiro et al., 2018; Silva et al., 2020), dentre muitas outras.

Neste capítulo, serão abordados os principais conceitos relacionados com a categorização de textos, bem como os principais tipos de problemas e aplicações de técnicas de AM na tarefa de categorização. Alguns problemas tradicionais, tais como análise de sentimento, filtragem de *spam* e sistemas de perguntas e respostas, estão detalhados nos Capítulos 25, 29 e 34, respectivamente.

23.1 Categorização de Textos e Aprendizado de Máquina

A categorização de textos é a tarefa de atribuir categorias (rótulos ou classes) a um documento escrito em linguagem natural (Sebastiani, 2002). A tarefa de classificar textos pode ser executada manualmente por um ser humano, uma vez que por meio de inspeção visual é possível interpretar o conteúdo do texto e escolher a categoria que o descreve melhor. No entanto, a categorização manual torna-se inviável quando existe um grande volume de documentos, o que demanda a aplicação de estratégias automatizadas.

Técnicas supervisionadas de AM são frequentemente empregadas para automatizar a tarefa de categorização de textos. A partir da observação de documentos rotulados, os algoritmos podem aprender diferentes padrões e associações entre segmentos de textos e rótulos. Esses padrões são então utilizados para categorizar documentos não rotulados, de acordo com os segmentos encontrados em seus conteúdos.

Formalmente, seja $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ um conjunto de n documentos ou amostras e $Y = \{y_1, y_2, ..., y_k\}$ um conjunto finito de categorias ou rótulos do problema, o conjunto de documentos rotulados que é apresentado ao algoritmo pode ser expresso como $\mathbf{D} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$, em que $\mathbf{x} \in \mathbf{X}$ e $\mathbf{y} \subseteq Y$. O objetivo no aprendizado supervisionado é encontrar uma função $H : \hat{f}(\mathbf{x}_i) \rightarrow \mathbf{y}_i$ (conhecida como modelo ou hipótese) que mapeia as associações presentes em $\mathbf{D}$ para predizer os rótulos de documentos não rotulados. Contudo, para que isso seja possível, é necessário que os documentos de textos sejam coletados, rotulados por especialistas, preparados e representados computacionalmente, conforme descrito nas próximas seções.

23.2 Coleta e Preparação do Texto

A primeira etapa no *pipeline* de um problema de categorização de textos é a coleta e rotulação manual dos documentos. Se, por exemplo, um documento de texto precisa ser extraído de uma página web, é necessário encontrar a parte de interesse do texto no código HTML da página. Em um problema de classificação de notícias, a parte de interesse do texto que pode ajudar a identificar a categoria à qual a notícia pertence é o título e o conteúdo. Portanto, deve-se procurar as marcações (*tags*) HTML que identificam onde essas partes de interesse estão inseridas. As marcações HTML que estão inseridas junto ao conteúdo, como as que definem estilos de formatação do texto (negrito, itálico, etc.), podem ser descartadas. Esse processo é conhecido como *parsing*. Alguns documentos podem ainda estar armazenados no formato de documento portátil (PDF – *portable document format*), linguagem de marcação extensível (XML – *extensible markup language*), valores separados por vírgulas (CSV – *comma separated values*), entre outros.

Na Figura 23.1, é apresentado um exemplo de documento de texto da base de dados Reuters-21578.[1] Nesse documento, as informações relevantes para serem usadas no processo de treinamento de um método de classificação são: (1) a categoria do documento que está sendo informada pela *tag* <D> que está dentro da *tag* <TOPICS>, (2) o título que está informado pela *tag* <TITLE> que está dentro da *tag* <TEXT> e (3) o conteúdo do texto que está informado pela *tag* <BODY> que está dentro da *tag* <TEXT>. Para automatizar a extração das informações de dentro das *tags* citadas, é possível usar um analisador sintático (*parser*).

[1] A base de dados Reuters-21578 está disponível em http://www.daviddlewis.com/. Acesso em: 7 jan. 2020.

```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="17858" NEWID=
   "1440">
<DATE> 4-MAR-1987 09:51:38.24</DATE>
<TOPICS><D>sugar</D></TOPICS>
<PLACES><D>uk</D><D>netherlands</D><D>denmark</D><D>west-germany</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS><D>ec</D></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN>
&#5; &#5; &#5;C T
&#22; &#22; &#1;f0234&#31;reute
b f BC-U.K.-INTERVENTION-BOA 03-04 0071</UNKNOWN>
<TEXT>&#2;
<TITLE>U.K. INTERVENTION BOARD DETAILS EC SUGAR SALES</TITLE>
<DATELINE> LONDON, March 4 - </DATELINE><BODY>A total 60,500 tonnes of current series
white sugar received export rebates of a maximum 43.147
European Currency Units (Ecus) per 100 kilos at today's
European Community (EC) tender, the U.K. Intervention Board
said.
Out of this, traders in the U.K. Received 43,500 tonnes, in
the Netherlands 12,000, in Denmark 4,000 and in West Germany
1,000 tonnes.
Earlier today, London and Paris traders said they expected
the subsidy for the current season whites campaign for licences
to end-July to be between 43.00 and 43.45 Ecus per 100 kilos.
They had also forecast today's total authorised sugar
tonnage export awards to be between 60,000 and 80,000 tonnes
versus 103,000 last week when the restitution was 43.699 Ecus.
REUTER
&#3;</BODY></TEXT>
</REUTERS>
```

FIGURA 23.1 Exemplo de um documento da base de dados Reuters-21578.

As etapas de coleta e rotulação manual dos documentos costumam ser as mais onerosas, pois cada forma de armazenamento requer uma maneira diferente de extração das informações desejadas para que, posteriormente, sejam estruturadas de forma a viabilizar seu uso por parte dos métodos de classificação. Ainda, é difícil fazer a limpeza dos documentos para eliminar as informações indesejadas, resolver problemas de codificação de caracteres e garantir a qualidade do texto extraído. Muitas vezes, a integridade do texto é comprometida no seu próprio armazenamento por falha humana ou por outros fatores (Weiss et al., 2004; Manning et al., 2009).

23.2.1 *Tokenização*

Depois que os documentos são coletados, uma etapa importante na estruturação da representação deles é a *tokenização*. Nesta etapa, é feita a segmentação do texto em termos (também chamados de *tokens*). Geralmente, um termo é um conjunto de caracteres alfanuméricos delimitado por caracteres delimitadores ("\n", "\t", "\r", entre outros), por espaços em branco ou por caracteres não alfanuméricos (ponto, vírgula, asterisco, entre outros). Nesse processo, pode ser definido um limite mínimo e máximo de caracteres para cada termo (Manning et al., 2009; Uysal e Gunal,

2014). Por exemplo, a *tokenização* do texto "*Esse#filme é muito bom! :D*", usando qualquer caractere não alfanumérico como delimitador e estipulando um limite mínimo de caracteres por termo igual a dois, resultaria no seguinte conjunto de termos: "*Esse*", "*filme*", "*muito*", "*bom*".

23.2.2 Técnicas de Pré-processamento

Muitas vezes, é necessário tratar os dados para padronizá-los e, assim, auxiliar os processos de treinamento e categorização. Nesta etapa, frequentemente são aplicadas as técnicas de pré-processamento apresentadas a seguir.

- *Conversão dos termos para letras minúsculas*: evita que duas palavras idênticas sejam consideradas diferentes pelo fato de suas letras estarem escritas em maiúsculo ou minúsculo (Uysal e Gunal, 2014).
- *Remoção de stopwords*: *stopwords* são termos muito comuns, tais como preposições, conjunções e artigos, que fornecem pouca ou nenhuma informação (Weiss et al., 2004). Portanto, eles podem ser removidos antes dos processos de treinamento e categorização.
- *Remoção de palavras raras*: remove os termos com baixa frequência no conjunto de documentos. A intuição é que termos que aparecem raras vezes no conjunto de treinamento não contribuem na identificação da categoria do documento e, portanto, podem ser descartados (Weiss et al., 2004).
- *Estemização*: reduz o termo ao seu radical (Uysal e Gunal, 2014). O processo de *estemização* varia de acordo com o idioma, sendo o algoritmo de Porter (1980) um dos mais usados para documentos escritos em língua inglesa. Para a língua portuguesa, um dos algoritmos mais utilizados na literatura é o RSLP (Orengo e Huyck, 2001). Para obter os radicais dos termos, os algoritmos de *estemização* geralmente removem os afixos e o final das palavras (mesmo que não sejam afixos) (Manning et al., 2009). Alguns exemplos são apresentados na Tabela 23.1.
- *Lematização*: reduz o termo à sua forma canônica, também conhecida como lema, por meio de análise morfológica (Manning et al., 2009). A lematização é mais complexa que a *estemização*, pois leva em conta a semântica do termo que está sendo analisado e, consequentemente, também possui maior custo computacional. Alguns exemplos são apresentados na Tabela 23.1.

Tabela 23.1 Exemplos de estemização e lematização de palavras da língua inglesa e da portuguesa

Palavra original	Estemização	Lematização
Palavras da língua inglesa		
studies	*studi*	*study*
university	*univers*	*university*
walking	*walk*	*walk*
Palavras da língua portuguesa		
empobrecendo	empobrec	empobrecer
felicíssimo	felic	feliz
segurados	segur	segurar

23.3 Representação Computacional

Os textos são escritos de forma que as ideias e informações possam ser facilmente compreendidas por seres humanos. No entanto, os computadores não são capazes de interpretar o conteúdo dos textos escritos em seu formato original. Tipicamente, técnicas de AM manipulam informações estruturadas, o que obriga que os textos escritos em linguagem natural passem por processos que os transformem em uma representação adequada, antes de poderem ser utilizados pelos métodos de AM.

Existem diversas formas de representar um documento de texto computacionalmente. Turian et al. (2010) agruparam as técnicas de representação de texto em três categorias: distributiva, agrupamento e distribuída. A *representação distributiva* é normalmente realizada pela construção de uma matriz de co-ocorrência, como a tradicional representação *bag of words*. Outras técnicas incluem o mapa semântico auto-organizável (Ritter e Kohonen, 1989), LSA (Landauer et al., 1998) e HAL (Lund e Burgess, 1996). A *representação baseada em agrupamento* busca induzir grupos sobre palavras, sendo que o principal trabalho relacionado é o agrupamento de Brown et al. (1992). Essas técnicas geralmente derivam um modelo de linguagem baseado em classes e, embora tenham apresentado bons resultados em tarefas específicas (tal como, reconhecimento de fala), ainda apresentam deficiências significativas, como a complexidade computacional durante a inferência estatística e a dependência de especialista para realizar a anotação manual de classes ou grupos. A *representação distribuída* é capaz de capturar similaridades semânticas ao analisar um grande volume de dados, representando unidades de texto em vetores densos, de tamanho fixo (Bengio, 2009).

Na representação distributiva, também conhecida como modelo espaço-vetorial, cada termo pode representar uma ou mais palavras do texto. Quando cada um deles representa um par de palavras do texto, eles são chamados de bigramas. Quando eles representam uma única palavra, são chamados de unigramas. A representação do documento que usa unigramas e é independente da sequência de palavras é chamada de *bag of words* (Sebastiani, 2002; Aggarwal, 2014).

Relembrando, seja $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ um conjunto com n documentos de textos, $T = \{t_1, t_2, ..., t_d\}$ o conjunto com d termos (atributos) que compõem um vocabulário predefinido e $Y = \{y_1, y_2, ..., y_k\}$ um conjunto finito de categorias ou rótulos do problema, o conjunto de documentos rotulados pode ser expresso como $\mathbf{D} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$, em que $\mathbf{x} \in \mathbf{X}$ e $\mathbf{y} \subseteq Y$. No modelo espaço vetorial, cada documento pode ser representado por uma matriz documento-termo, onde cada posição contém um peso $w(t_i, \mathbf{x}_j)$ associado a cada termo, conforme mostra a Tabela 23.2.

Tabela 23.2 Representação dos documentos usando o modelo espaço-vetorial

Documentos	t_1	t_2	t_3	...	t_d	Categoria
$\mathbf{x}_1$	$w(t_1, \mathbf{x}_1)$	$w(t_2, \mathbf{x}_1)$	$w(t_3, \mathbf{x}_1)$	...	$w(t_d, \mathbf{x}_1)$	$\mathbf{y}_1$
$\mathbf{x}_2$	$w(t_1, \mathbf{x}_2)$	$w(t_2, \mathbf{x}_2)$	$w(t_3, \mathbf{x}_2)$	...	$w(t_d, \mathbf{x}_2)$	$\mathbf{y}_2$
$\mathbf{x}_3$	$w(t_1, \mathbf{x}_3)$	$w(t_2, \mathbf{x}_3)$	$w(t_3, \mathbf{x}_3)$	...	$w(t_d, \mathbf{x}_3)$	$\mathbf{y}_3$
⋮	⋮	⋮	⋮	⋱	⋮	⋮
$\mathbf{x}_n$	$w(t_1, \mathbf{x}_n)$	$w(t_2, \mathbf{x}_n)$	$w(t_3, \mathbf{x}_n)$	...	$w(t_d, \mathbf{x}_n)$	$\mathbf{y}_n$

Os valores usados para representar os termos de um documento ($w(t_i, \mathbf{x})$) geralmente são baseados na frequência da ocorrência de t_i em $\mathbf{x}$ e podem variar dependendo da estratégia de atribuição de pesos aplicada, sendo que as mais comuns são apresentadas a seguir.

- *Binária*: neste tipo de representação, o termo recebe o valor um, caso apareça no documento, e zero, caso não apareça. Portanto, a frequência com que o termo aparece não é considerada. Matematicamente, o peso binário pode ser calculado por:

$$w(t_i, \mathbf{x}) = \begin{cases} 1, & \text{se } t_i \text{ aparece em } \mathbf{x}; \\ 0, & \text{se } t_i \text{ não aparece em } \mathbf{x}. \end{cases} \quad \textbf{(23.1)}$$

- *Frequência do termo (*TF – *term frequency)*: os pesos dos termos correspondem à quantidade de vezes que o termo apareceu no documento.
- *Frequência do termo-frequência inversa dos documentos (*TF-IDF – *term frequency-inverse document frequency)*: o peso do termo é igual ao produto da sua frequência pela frequência inversa dos documentos (IDF – *inverse document frequency*) (Salton et al., 1975). A IDF mede a quantidade de informação que um termo carrega, o que depende do quão frequente ele é no conjunto de dados de treinamento. Um alto valor de TF-IDF é obtido quando o termo tem uma alta frequência no documento que está sendo avaliado e uma baixa frequência no conjunto de dados de treinamento (Wilbur e Kim, 2009; Rennie et al., 2003). Para calcular o peso TF-IDF de um termo t_i qualquer, pode ser aplicada a seguinte equação:

$$\begin{aligned} w(t_i, \mathbf{x}) &= \overline{TF}(t_i, \mathbf{x}) \times IDF_{t_i} \\ &= \log(1 + TF(t_i, \mathbf{x})) \times \log\left(\frac{n+1}{DF_{t_i}+1}\right), \end{aligned} \tag{23.2}$$

em que $TF(t_i, \mathbf{x})$ é a frequência do termo t_i no documento $\mathbf{x}$, n é a quantidade de documentos no conjunto de treinamento e DF_{t_i} é o número de documentos de treinamento que contém o termo t_i (Wilbur e Kim, 2009). A normalização L2 (também conhecida como normalização euclidiana) pode ser aplicada no peso TF-IDF obtido na Equação 23.2 (Rennie et al., 2003):

$$\hat{w}(t_i, \mathbf{x}) = \frac{w(t_i, \mathbf{x})}{\| w(:, \mathbf{x}) \|_2} \tag{23.3}$$

A escolha do esquema de peso mais adequado depende do problema e do método de classificação que será empregado, pois cada um deles pode exigir um esquema de peso diferente. A Figura 23.2 ilustra um exemplo da representação usando as frequências dos termos (TF). Apesar de simples, o modelo espaço-vetorial obtém bons resultados na maioria das aplicações (Silva, 2017; Weiss et al., 2004).

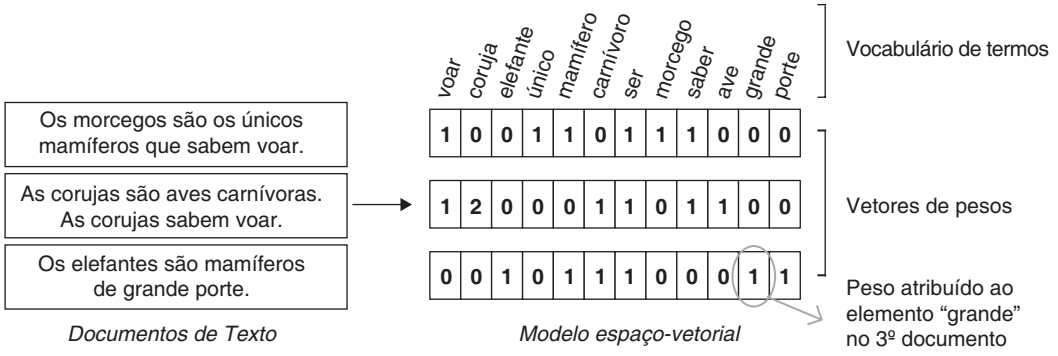

FIGURA 23.2 Transformação dos documentos de texto para o modelo espaço-vetorial (Bittencourt, 2020).

Apesar de ser amplamente utilizado em representação de textos, o modelo *bag of words* possui algumas deficiências bem conhecidas, discutidas a seguir.

A quantidade de dimensões pode ser muito alta

Para a maioria dos algoritmos de AM, um dos problemas com o uso da *bag of words* é que a quantidade de dimensões pode ser alta, pois corresponde ao tamanho do vocabulário. Isso pode ocasionar superajuste aos dados e aumento considerável no custo computacional de treinamento. Além disso, como cada amostra contém apenas uma fração das palavras do vocabulário, existe o problema da alta esparsidade. No exemplo artificialmente simples ilustrado na Figura 23.2, com 3 amostras e 12 termos, isso já pode ser observado. A maioria dos termos não está presente na maioria das amostras. Esse fato se agrava em problemas reais que podem contar com milhões de documentos e com milhares de termos no vocabulário.

A informação sobre a posição relativa dos termos no texto é perdida

Para alguns problemas de categorização de textos, a simples ocorrência de determinadas palavras pode ser suficiente para classificar um documento corretamente. Contudo, para problemas como análise de opinião, a ordem das palavras pode ser muito importante. As sentenças "O vírus destruiu o sistema imunológico do paciente" e "O sistema imunológico do paciente destruiu o vírus" contém exatamente as mesmas palavras, mas têm significados diferentes. Assim, saber somente quais palavras ocorrem em cada frase pode não ser suficiente para que ela seja classificada corretamente.

Sinônimos, gírias, abreviações e erros de escrita

Abreviações e gírias são muito usadas, especialmente no caso de textos curtos, como nas mensagens postadas em redes sociais e enviadas por mensageiros instantâneos. Por exemplo, a mensagem "*dz ne1 knw h2 ripair dis terrible LPT?*" traduzida para o inglês formal equivaleria a: "*Does any one know how to repair this terrible printer?*". O grande problema com essas palavras é que elas representam diferentes maneiras de se referir a mesma coisa, o que corresponde ao mesmo problema dos sinônimos. No exemplo, "*LTP*" equivale a "*printer*" (impressora). Mesmo que "*LTP*" fizesse parte do vocabulário, o algoritmo de aprendizado não teria como associar "*LTP*" com "*printer*". Um modelo de classificação que tivesse sido treinado com a amostra "Esse filme é sensacional" como uma avaliação positiva, não levaria em consideração o fato de "sensacional" ser sinônimo de "incrível" ao ser usado para classificar "Esse filme é incrível", pois essas duas palavras são atributos diferentes que para o modelo não têm nenhuma relação.

Ambiguidade

Algumas palavras podem ter vários significados diferentes. Nas frases "Meu caro amigo" e "Achei aquele celular muito caro", as ocorrências da palavra "caro" denotam opiniões positiva e negativa, respectivamente. Geralmente, o significado correto pode ser determinado pelo contexto onde a palavra ocorre. Entretanto, com o uso de *bag of words*, o contexto é descartado e essa distinção não pode ser feita.

23.3.1 Representação Computacional de Textos Curtos e Ruidosos

Na representação computacional de mensagens curtas, os problemas da *bag of words* são agravados. O limite de caracteres imposto para o tamanho dessas mensagens faz com que o processo de categorização seja particularmente mais difícil. Menos caracteres equivale a menos termos, o que faz com que cada amostra usada para treinar os algoritmos de classificação contenha menos atributos (Silva et al., 2017b).

O uso dos *smartphones* para acessar as redes sociais, como alternativa ao computador, amplifica esse problema. Além do texto ter que ser reduzido, geralmente as pessoas escrevem rapidamente, utilizando o teclado diminuto do *smartphone* com apenas dois dedos, sendo que as mãos são frequentemente utilizadas para segurar o celular e digitar o texto ao mesmo tempo. Em virtude da união desses fatores, os usuários criaram uma linguagem para comunicar suas mensagens com a maior brevidade possível. Embora essa brevidade permita que as mensagens transmitam mais informações usando menos caracteres, elas dificultam a representação computacional em razão da falta de padronização.

Essa linguagem extremamente coloquial contém uma quantidade alta de repetições, acrônimos e de neologismos. Acrônimos são termos que se formam pela junção das primeiras letras ou das sílabas iniciais de um grupo de termos. Por exemplo, os termos *"omg"*, *"brb"* e *"lol"* são frequentemente usados no lugar de *"oh my god"*, *"be right back"* e *"laugh out loud"*, respectivamente. Outro recurso muito usado nessa nova linguagem é a ortografia fonética, ou seja, palavras escritas de maneira errada cuja pronúncia é similar à pronúncia da palavra correta, como, por exemplo, *"tryna"* no lugar de *"trying to"* (Almeida et al., 2016).

Nesse tipo de mensagem, também existe pouca consideração pelo uso adequado de letras maiúsculas e pontuações. Letras maiúsculas podem sinalizar um nome próprio ou o fim da sentença, mas também podem ser usadas para algo tão arbitrário quanto enfatizar um determinado segmento. A pontuação pode determinar os limites da sentença, mas também pode ser empregada para criar *emoticons*, como ":(" e ";-)", que são usados para expressar emoções, sentimentos ou ideias.

Diversas técnicas de pré-processamento vêm sendo propostas e utilizadas para melhorar a representação computacional de textos curtos e ruidosos, sendo que as mais tradicionais envolvem normalização léxica, indexação semântica e desambiguação de sentido das palavras.

Normalização léxica

Tradicionalmente, para lidar com os problemas das gírias, abreviações e erros de escrita, são utilizadas técnicas de normalização léxica para traduzir variantes de palavras e expressões para sua forma canônica. No caso de mensagens curtas e ruidosas, embora haja similaridade com a tarefa de correção ortográfica, a tarefa de normalização léxica é mais desafiadora porque em muitos casos as palavras são intencionalmente escritas de maneira errada, em função do limite de caracteres ou para adotar o estilo característico do meio utilizado. Por exemplo, substituir "*b4*" por "*before*" está além da capacidade de um simples corretor ortográfico, assim como é o caso da repetição de letras para dar ênfase como em "*goooood*" (Clark e Araki, 2011; Almeida et al., 2016). Para fazer essa tradução, são utilizados dicionários conhecidos como *Lingo*, como, por exemplo, o dicionário `NoSlang`.[2] Um exemplo é apresentado na Tabela 23.3.

Dicionários semânticos e desambiguação de sentido da palavra

Para abordar o problema dos sinônimos e da polissemia, tradicionalmente se faz uso de dicionários semânticos e algoritmos de desambiguação do sentido da palavra. Dicionários semânticos como o `LDB WordNet`[3] relacionam palavras com os seus diferentes sentidos. Desambiguação de sentido de uma palavra (do inglês *word sense disambiguation*) é a atividade de encontrar, dada a ocorrência em uma palavra ambígua no texto, o sentido específico daquela ocorrência de acordo com o contexto em que ela ocorre (Almeida et al., 2016; Silva et al., 2017b). Um exemplo é apresentado na Tabela 23.3.

Tabela 23.3 Exemplo de normalização léxica, indexação semântica e desambiguação

Texto original	*U shud try d nu device... It iz gr8*
Normalização léxica	*you should try the new device it be great*
Indexação semântica	*you should attempt effort endeavor endeavour try the new applience device gadget instrument gimmick twist it be great*
Desambiguação	*you should attempt the new device it be great*

23.3.2 Representação Distribuída

A indexação semântica e expansão de amostras, embora ajudem em diversas aplicações, também podem piorar o resultado, quando a desambiguação não é feita, ou é feita de maneira imperfeita (Lochter et al., 2016). Outro problema é que esses métodos possuem alto custo computacional, sendo que cada palavra da amostra, junto com seu contexto, devem ser analisados. O maior problema, entretanto, é que todos esses métodos são dependentes de dicionários. As línguas evoluem com o tempo, e cada vez mais frequentemente surgem novas palavras ou novos significados são atribuídos a palavras existentes. Nas redes sociais, novos termos surgem e se espalham rapidamente, passando a fazer parte do "dialeto" utilizado. Além disso, para classificar textos em diferentes línguas são necessários diferentes dicionários, sendo que as atualizações constantes podem ter um custo proibitivo em cenários de aplicações reais.

Para contornar essas limitações, o uso de *representações vetoriais distribuídas de palavras e parágrafos* é uma alternativa que dispensa o emprego de dicionários, e tem o potencial de resolver os principais problemas relacionados com as representações baseadas em *bag of words*. As representações distribuídas são geradas de maneira não supervisionada a partir de um grande volume de texto, sendo que nesses modelos cada palavra corresponde a um vetor de

[2] O dicionário `NoSlang` está disponível em: http://www.noslang.com/dictionary/full/. Acesso em: 7 jan. 2020.

[3] O `LDB WordNet` está disponível em: https://wordnet.princeton.edu/. Acesso em: 9 jan. 2020.

baixa dimensionalidade. Palavras que aparecem em contextos similares no texto corresponderão a vetores próximos nesse espaço. Dessa maneira, a similaridade semântica entre as palavras é preservada.

Como a similaridade semântica é preservada, algoritmos de aprendizado podem aprender com palavras diferentes, mas com significados similares, como no exemplo de "incrível" e "sensacional". Pelo fato de os modelos serem gerados automaticamente a partir de uma grande quantidade de texto, eles podem incorporar gírias e abreviações sem a constante necessidade de atualizar dicionários. Por exemplo, é esperado que o termo "*hj*" seja representado por um vetor não muito distante do vetor que representa a palavra "hoje".

Na representação distribuída, as representações encontradas para cada termo são conhecidas como *word embeddings*, onde cada dimensão pode mapear características semânticas ou sintáticas dependendo do treinamento realizado e do corpo de texto empregado. Na literatura, tais representações são induzidas por várias técnicas, sendo a modelagem baseada em redes neurais artificiais a mais conhecida (Mikolov et al., 2011; Socher, 2015).

Um exemplo de arquitetura de rede neural artificial para induzir *word embeddings* é apresentado na Figura 23.3. Nela, as amostras de entrada são conjuntos de três palavras (termos 1, 2 e 3) que existem no dicionário conhecido, enquanto a saída (termo 4) é a predição da palavra subsequente mais provável às três palavras anteriores. Por exemplo, para a amostra de entrada *"eu gosto de"* em uma revista de viagens, é esperado que a palavra subsequente *"viajar"* seja mais provável de ocorrer do que a palavra *"sofrer"*.

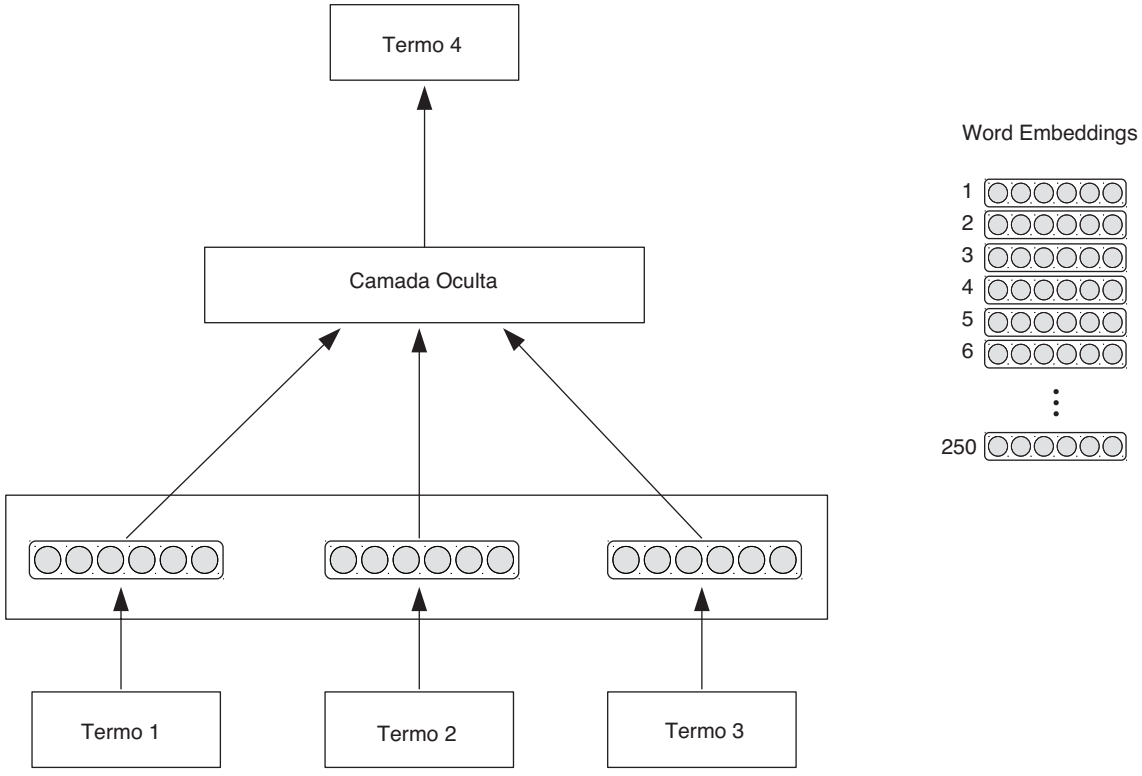

FIGURA 23.3 Exemplo de rede neural artificial para indução de representação distribuída (Lochter et al., 2018a).

O funcionamento da rede consiste em atualizar a *word embedding* de cada um dos três termos baseado no erro calculado entre o quarto termo esperado como resposta e o termo com maior probabilidade obtido pela rede. O erro calculado na rede é propagado de volta para as *word embeddings* de entrada, as quais são atualizadas e usadas posteriormente, quando novas amostras forem apresentadas e contiverem algumas das palavras vistas anteriormente pela rede.

A representação distribuída tem conduzido a resultados promissores, principalmente quando aplicadas a uma amostra de texto completa, em vez de apenas a palavras (Socher, 2015; Kusner et al., 2015; Wu et al., 2017). As técnicas mais usadas para gerar representações distribuídas de textos são Word2vec (Mikolov et al., 2013), GloVe (Socher, 2015) e FastText (Bojanowski et al., 2017).

23.4 Métodos de Classificação

Uma variedade de métodos têm sido empregados na categorização de textos, que se diferem pela estratégia empregada para se obter a função hipótese *H*. As principais estratégias são descritas a seguir (Silva, 2017; Sebastiani, 2002):

- Métodos baseados em distâncias: consideram a proximidade entre os documentos para realizar as predições. O método dos *k*-vizinhos mais próximos (Cover e Hart, 1967) é o mais conhecido.
- Métodos probabilísticos: se baseiam na probabilidade de o documento pertencer a cada uma das classes possíveis do problema. O Bayes ingênuo (NB – *naive* Bayes), as redes bayesianas (McCallum e Nigam, 1998) e o MDLText (Silva et al., 2017c; Freitas et al., 2019) são exemplos de métodos probabilísticos.
- Métodos baseados em árvores de decisão: constituídos de métodos que dividem um problema complexo em subproblemas mais simples, sob uma estrutura de árvore. As árvores de classificação e regressão (CART – *classification and regression trees*) (Breiman et al., 1984) e o C4.5 (Quinlan, 1993) são os métodos mais tradicionais baseados em árvore de decisão.
- Métodos baseados em otimização: a hipótese é encontrada a partir da otimização de alguma função que avalia a capacidade de predição. As máquinas de vetores de suporte (SVM – *support vector machines*) (Cortes e Vapnik, 1995) e as redes neurais artificias (Haykin, 1999a) são duas técnicas muito empregadas com sucesso em diversas aplicações.
- Métodos *ensemble*: treinam diferentes classificadores para a mesma tarefa de classificação e combinam os julgamentos individuais desses classificadores para gerar a predição final (Sebastiani, 2002). Floresta aleatória e o reforço adaptativo (AdaBoost) (Freund e Schapire, 1996) são exemplos bastante utilizados.

23.5 Tipos de Problemas de Categorização de Textos

As tarefas de categorização podem ser distinguidas pela quantidade de categorias que são atribuídas para cada documento. Em algumas aplicações, a tarefa de classificação é denominada *monorrótulo*, pois apenas uma categoria $y_i \in Y$ é atribuída aos documentos de texto. Em outras situações, os documentos de textos podem ser vinculados a um subconjunto de rótulos $\mathbf{y}_i \subseteq Y$, ou seja, múltiplos rótulos podem ser atribuídos ao mesmo documento. Nestes casos, a tarefa de classificação é denominada *multirrótulo*.

23.5.1 Classificação Monorrótulo

A classificação monorrótulo é a abordagem padrão no aprendizado supervisionado. Quando o problema tem apenas duas classes ($|Y| = 2$), ele é chamado de classificação *binária*. Filtragem de *spam* é um exemplo desse tipo de problema, já que existem duas possibilidades de rótulos para as mensagens de *e-mail*: "*spam*" e "não *spam*".

Em casos onde o problema de classificação contém mais que duas classes ($|Y| > 2$), ele recebe o nome de classificação *multiclasse*. Um exemplo é a categorização dos documentos de uma empresa de acordo com o setor responsável por ele, por exemplo: "financeiro", "produção" ou "marketing".

23.5.2 Classificação Multirrótulo

Muitos problemas de categorização dependem de uma rotulação múltipla. Para visualizar as diferenças entre as classificações monorrótulo e multirrótulo, a Figura 23.4 ilustra uma notícia categorizada seguindo esses dois contextos. Um texto pode abordar simultaneamente diversos assuntos e a categorização multirrótulo permite extrair mais conceitos e informações, o que é útil em uma infinidade de aplicações. Uma mensagem de *e-mail*, por exemplo, pode ser categorizada de acordo com os diversos assuntos contidos em seu conteúdo. No diagnóstico médico, o paciente pode apresentar diversas doenças simultaneamente. Um filme, por sua vez, pode ser rotulado com um ou mais gêneros diferentes.

Existe ainda um subconjunto de problemas de classificação multirrótulo, denominado hierárquico, no qual as classes do problema se encontram organizadas em uma estrutura de hierarquia. Um exemplo clássico é a organização das notícias de jornal, cujos tópicos podem estar divididos em subcategorias, como, por exemplo: "esportes" $\rightarrow$ "futebol".

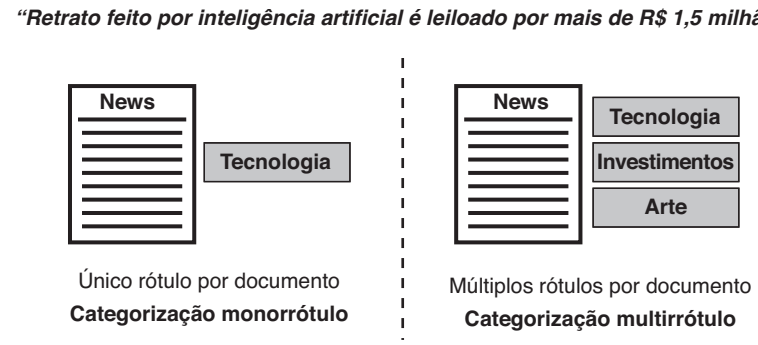

FIGURA 23.4 Tipos de problemas de categorização de textos (Bittencourt et al., 2019).

Embora existam diversos problemas textuais que dependem da classificação multirrótulo, os estudos sobre este tipo de classificação estão bem menos presentes na literatura do que a classificação monorrótulo. Alguns trabalhos manipulam o problema de classificação multirrótulo como um conjunto de problemas de classificação monorrótulo e ignoram as características intrínsecas da rotulação múltipla. Como mais de um rótulo pode ser atribuído aos textos, a classificação multirrótulo se torna mais desafiadora do que a monorrótulo, já que o número de rótulos de predição é uma nova incógnita para o classificador. Além disso, existem mais relações entre os dados de observações e rótulos, o que torna a escalabilidade uma característica importante para a escolha da abordagem de categorização mais apropriada. Discussões acerca de estratégias para tratar problemas multirrótulo e hierárquicos são apresentadas nos Capítulos 19 e 20, respectivamente.

23.6 Aprendizado *Online* e *Offline*

A técnica mais apropriada para um problema de categorização pode ser definida também de acordo com a forma com que a etapa de treinamento deve ser realizada. Diversos métodos requerem que todos os documentos rotulados sejam apresentados simultaneamente, em um processo único de treinamento conhecido como treinamento em *lote* ou *offline*. Esses métodos não são capazes de incorporar novas informações no modelo de predição após a etapa de treinamento, o que exige que ela seja refeita desde o ponto inicial quando novos documentos rotulados se tornarem disponíveis para o treinamento (Silva et al., 2017c).

A categorização de textos costuma manipular problemas com muitos documentos e com altas dimensões no espaço de atributos, portanto, carregar todos os documentos em memória para o treinamento em lote nem sempre é possível. Em muitos problemas reais, os documentos textuais não estão todos disponíveis para o treinamento do classificador ou os conceitos das classes podem mudar com o tempo. Nesses cenários, os métodos de classificação devem suportar o aprendizado *incremental* ou *online*.

No aprendizado *online*, o classificador é construído por meio de um pequeno conjunto de documentos de treinamento e pode ser atualizado incrementalmente conforme novos documentos se tornam disponíveis. O algoritmo Perceptron (Wiener et al., 1995) é um método muito conhecido que permite o aprendizado incremental. Outros métodos de aprendizado *online* bastante conhecidos são naive Bayes, gradiente descendente estocástico (SGD – *stochastic gradient descent*) (Zhang e Zhou, 2014), passivo-agressivo (PA) (Crammer et al., 2006) e o MDLText (Silva et al., 2017c; Freitas et al., 2019). Apesar de métodos de aprendizado *online* terem vantagens em problemas dinâmicos ou de larga escala, métodos de aprendizado *offline* costumam obter melhores resultados quando esses podem ser aplicados (Crammer et al., 2012).

23.7 Seleção de Atributos

Uma etapa bastante empregada no processo de categorização de textos é a seleção de atributos (termos). Seu principal objetivo é eliminar termos pouco informativos, redundantes ou até mesmo ruidosos. Em alguns cenários, ela pode melhorar o desempenho da classificação, aumentando a capacidade de generalização, diminuindo o tempo de aprendizado e simplificando o modelo de predição (Bolón-Canedo et al., 2015).

As três principais categorias de métodos de seleção de atributos são: filtros, métodos *wrappers* e métodos *embedded*. Retomando a apresentação no Capítulo 3, tem-se:

- *Filtros*: realizam a seleção de atributos antes do processo de treinamento e são independentes do método de classificação. Geralmente, os filtros utilizam informações estatísticas do conjunto de dados de treinamento para determinar a relevância dos atributos. Depois disso, pode ser selecionado o conjunto dos k atributos mais informativos, enquanto os outros são descartados. Outra alternativa é selecionar apenas os atributos que possuem um valor de relevância maior do que um determinado limiar (Liu e Motoda, 1998).
- *Métodos wrappers*: utilizam informações sobre o desempenho do método de aprendizado como parte de sua função de avaliação da relevância dos atributos. Eles criam diferentes subconjuntos de atributos, avaliam o desempenho do método de aprendizado usando cada um dos subconjuntos e, geralmente, selecionam aquele que melhora os resultados. Os métodos *wrappers* podem utilizar estratégias de busca para explorar o espaço de todos os possíveis subconjuntos de atributos, pois normalmente a quantidade de combinações é muito extensa, o que inviabiliza uma busca exaustiva (Liu e Motoda, 1998; Chandrashekar e Sahin, 2014).
- *Métodos embedded*: aplicam a seleção de atributos como parte do modelo de predição e são específicos para os métodos de aprendizado ao qual são embutidos. Por exemplo, um método *embedded* pode ser embutido ao modelo de predição como um parâmetro de regularização que atribui pesos aos atributos. Nesse caso, os atributos irrelevantes ou redundantes podem receber um peso zero para que não exerçam nenhuma influência na predição (Chandrashekar e Sahin, 2014).

Diversos métodos de seleção de atributos são descritos em Liu e Motoda (1998), Sebastiani (2002), Chandrashekar e Sahin (2014) e Bolón-Canedo et al. (2015). Os métodos de seleção baseados em filtros são os mais simples e os mais escaláveis. Já os métodos *wrappers* e *embedded* geralmente demandam um alto custo computacional e, por isso, são menos indicados para problemas de categorização de texto reais e de grande escala.

A maioria dos métodos de seleção de atributos é usada em cenários de aprendizado *offline*. A utilização de métodos de seleção de atributos para cenários de aprendizado *online* ainda é um desafio, pois esse processo requer um número suficiente de exemplos para obter um desempenho estável e aceitável. Além disso, os poucos métodos de seleção de atributos *online* disponíveis na literatura geralmente consideram que o número de exemplos de treinamento é constante. Ainda, em problemas reais e *online*, sempre que um novo exemplo é apresentado para um método de aprendizado, é necessário considerar a possibilidade de (1) incluir um atributo que foi descartado anteriormente, (2) selecionar novos atributos que estão presentes no novo exemplo e que não haviam ocorrido em nenhum exemplo anterior e (3) remover atributos que foram selecionados anteriormente, mas que depois perderam a relevância (Wu et al., 2010, 2013; Bolón-Canedo et al., 2015; Tang et al., 2014).

23.8 Considerações Finais

Este capítulo introduziu os principais conceitos relacionados com a categorização e representação computacional de textos. Inicialmente, foi discutido o processo de coleta e *tokenização* de documentos e foram apresentadas as principais técnicas de pré-processamento empregadas para facilitar o processo de categorização. Além disso, também foram brevemente descritas as técnicas de normalização léxica, indexação semântica e desambiguação, comumente aplicadas para minimizar problemas de representação de textos curtos e ruidosos.

Também, foi discutido neste capítulo como pode ser construído o modelo espaço-vetorial para a representação estruturada dos documentos e foram apresentadas as principais estratégias de atribuição de pesos para os termos: TF, TF-IDF e binária. Como o desempenho de alguns métodos de categorização de texto pode ser bastante afetado pela estratégia escolhida, é recomendado realizar uma busca em grade (*grid search*) para selecionar a mais adequada ao método e à aplicação. Adicionalmente, foi introduzida a representação distribuída de termos como uma boa alternativa à tradicional *bag of words*.

Por fim, foram apresentados os principais tipos de métodos de seleção de atributos e os tipos de problemas e aprendizado em categorização de textos. A abordagem padrão de classificação em AM considera que as classes são mutuamente exclusivas, ou seja, diferentes classes não podem ocorrer simultaneamente para um mesmo documento. Entretanto, uma grande gama de problemas ignora esta restrição e depende de abordagens específicas, capazes de prever não apenas um, mas múltiplos rótulos para um mesmo documento.