Realização

SBC

WebMedia2022

Minicursos | XXVIII Simpósio Brasileiro de Sistemas Multimídia e Web

07 a 11 de Novembro de 2022

Curitiba| PR| Brazil

ORGANIZADORES

Débora C. Muchaluat Saade (UFF), Rodrigo Minetto (UTFPR),
Roberto Willrich (UFSC), Thiago Henrique Silva (UTFPR) e
Leyza Baldo Dorini (UTFPR)

Patrocínio

Cooperação

Organização

XXVIII Simpósio Brasileiro de Sistemas Multimídia e Web

De 7 a 11 de novembro de 2022

Curitiba, Brasil

LIVRO DE MINICURSOS

Organizadores

Débora C. Muchaluat Saade (UFF)
Rodrigo Minetto (UTFPR)
Roberto Willrich (UFSC)
Thiago Henrique Silva (UTFPR)
Leyza Baldo Dorini (UTFPR)

Realização

Sociedade Brasileira de Computação – SBC

Em cooperação com

ACM/SIGWEB e ACM/SIGMM

Organização

Universidade Tecnológica Federal do Paraná (UTFPR)

Dados Internacionais de Catalogação na Publicação

Simpósio Brasileiro de Sistemas Multimídia e Web (28. : 7-11 nov. 2022 : Curitiba - PR)

Minicursos do XXVIII Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia) [recurso eletrônico] / organização Débora C. Muchaluat Saade... [et al.]. -- Curitiba, PR : Universidade Tecnológica Federal do Paraná: Sociedade Brasileira de Computação, 2022.

1 recurso eletrônico (195 p.) : il. PDF; 7,62MB

ISBN 9788576695127

Inclui bibliografias.

Modo de acesso: World Wide Web

Título extraído da tela de título (visualizado em 23 set. 2022)

1. Sistemas multimídia - Congressos. 2. World Wide Web (Sistema de recuperação da informação). 3. Redes sociais on-line. 4. Multimídia interativa. 5. Hipermídia. I. Saade, Débora Cristina Muchaluat. II. Universidade Tecnológica Federal do Paraná. III. Sociedade Brasileira de Computação. IV. Título.

CDD: ed. 23 -- 006.7

Departamento de Bibliotecas da Universidade Tecnológica Federal do Paraná
Bibliotecário: Adriano Lopes, CRB-9/1429

Prefácio

Tradicionalmente, o Simpósio Brasileiro de Sistemas Multimídia e Web oferece à sua comunidade minicursos de curta duração relacionados a temas que norteiam os últimos avanços na área de multimídia e web. Este também é o caso do XXVIII Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia 2022), onde os minicursos têm a duração de quatro horas e permitem que participantes recebam informações sobre novas tecnologias e tópicos atuais de pesquisa em áreas correlatas ao evento. Assim, minicursos aparecem como uma excelente oportunidade para familiarização dos congressistas com novos temas de pesquisa que podem vir a ser úteis em suas vidas profissionais. Na edição de 2022, além de minicursos, o WebMedia também inovou com a chamada de submissão de tutoriais, que foram ministrados com duração de duas horas e foram publicados como artigos curtos nos anais estendidos do evento.

Para o Webmedia 2022, o processo de seleção de minicursos foi feito a partir de várias chamadas públicas divulgadas na lista eletrônica de emails da SBC, bem como ampla divulgação no site oficial do evento e em redes sociais. Foram recebidas oito propostas de minicursos, avaliadas por um comitê composto de professores com conhecimento nos temas abordados. Cada minicurso foi avaliado por três avaliadores, que pontuaram notas para quesitos como relevância para o evento, expectativa de público, atualidade e conteúdo de cada minicurso. Ao final, quatro propostas foram selecionadas, cujos conteúdos abordados constituem os capítulos deste livro. O processo de seleção de tutoriais foi similar, tendo sido recebidas quatro propostas, das quais duas foram aceitas para publicação e apresentação durante o evento.

O Capítulo 1, intitulado *Identificação de Câmaras de Eco em Redes Sociais através de Detecção de Comunidade em Redes Complexas: Ferramentas, Tendências e Desafios*, aborda os principais algoritmos para a caracterização estrutural e técnicas que auxiliam na detecção de câmaras de eco em redes sociais. A câmara de eco é um fenômeno relacionado à tendência de usuários de interagirem com outros usuários em grupos homogêneos e com ideias e opiniões semelhantes, fomentando ambientes propícios ao discurso de ódio e à propagação de notícias falsas (*fake news*). O capítulo se concentra em abordagens de descoberta de comunidades sobre um gráfico de topologia criado de acordo com a difusão de informações em redes sociais. Detalham-se também algoritmos de caracterização de redes complexas e os índices de desempenho dessas abordagens. Por fim, são discutidos os desafios e projetos de pesquisa que focam no estudo de câmaras de eco em redes sociais online.

O Capítulo 2, *Processamento de Linguagem Natural em Textos de Mídias Sociais: Fundamentos, Ferramentas e Aplicações*, tem como objetivo principal apresentar fundamentos e tecnologias na área de Processamento de Linguagem Natural (PLN - ou NLP, do acrônimo em Inglês de *Natural Language Processing*) para o desenvolvimento de aplicações por meio da exploração de textos de mídias sociais escritos em língua inglesa. O capítulo apresenta diversas técnicas de NLP para construir e executar pipelines de NLP, incluindo as etapas de pré-processamento, representação, modelagem, extração de conhecimento, compreensão semântica e emocional, a partir de textos de mídias sociais. Também

apresenta possíveis aplicações que podem se beneficiar do conhecimento extraído de tais textos.

O Capítulo 3 é intitulado *Polarização em Redes Sociais: Conceitos, Aplicações e Desafios*. Este capítulo apresenta o fluxo de coleta de dados sobre polarização, seu processamento, análises e extração de conhecimento. É dado enfoque especial em uma proposta de taxonomia para métricas de polarização em redes sociais. Também são discutidos desafios e oportunidades encontrados na área de polarização em redes sociais, assim como seus impactos na sociedade e na Internet.

O Capítulo 4, intitulado *Geração de Séries Temporais Utilizando Redes Generativas Adversárias: da Teoria à Prática*, discute a geração de dados sintéticos a partir dos dados originais, preservando suas características e mantendo sua privacidade. Como as séries temporais são um tipo de dados dependentes do tempo, podem representar desafios adicionais para esses modelos. O capítulo apresenta as Redes Generativas Adversárias (*Generative Adversarial Networks* - GANs), um framework baseado em *deep learning* para treinamento de modelos generativos.

Agradecemos aos membros do comitê de minicursos e tutoriais pelo trabalho na avaliação e seleção das propostas aceitas e esperamos que este livro seja útil para todos aqueles interessados nos temas abordados.

Curitiba, novembro de 2022.

Débora Christina Muchaluat Saade (UFF)
Rodrigo Minetto (UTFPR)
Coordenadores dos Minicursos do WebMedia 2022

XXVIII Simpósio Brasileiro de Sistemas Multimídia e Web

7 a 11 de novembro de 2022
Curitiba, Brasil

Coordenação Geral

Thiago Henrique Silva (UTFPR) – *Coordenador Geral*
Leyza Baldo Dorini (UTFPR) – *Coordenadora Geral*
Jussara M. Almeida (UFMG) – *Coordenadora do Comitê de Programa*
Humberto Torres Marques-Neto (PUC Minas) – *Coordenador do Comitê de Programa*

Coordenador do IV Concurso de Teses e Dissertações (CTD)

Leonardo Rocha (UFSJ)
Windson Viana de Carvalho (UFC)

Coordenadores do II Concurso de Trabalhos de Iniciação Científica (CTIC)

Eduardo Barrére (UFJF)
Michele A. Brandão (IFMG)

Coordenadores do XXI Workshop de Ferramentas e Aplicações e Prêmio LF

Alan Guedes (UCL, UK)
André Santanchè (Unicamp)
Felipe Domingos da Cunha (PUC Minas)
Luiz Celso Gomes Jr (UTFPR)

Coordenador de Publicação

Roberto Willrich (UFSC)

Coordenação de Patrocínios e Contatos Institucionais

Adriano César Machado Pereira (UFMG)
Carlos André G. Ferraz (UFPE)

Coordenação de Palestras e Painéis

Alexandre Graeml (UTFPR)
Myriam R. de B. da Silva Delgado (UTFPR)
Pedro O.S Vaz-de-Melo (UFMG)

Coordenação de Minicursos e Tutoriais

Débora C. Muchaluat Saade (UFF)
Rodrigo Minetto (UTFPR)

Coordenação de Competições

Diego Roberto Colombo Dias (UFSJ)
Flavio Vinicius Diniz de Figueiredo (UFMG)
Vinícius Fernandes Soares Mota (UFES)

Comissão de Apoio Local

Anelise Munaretto (UTFPR)
Carmem Hara (UFPR)
David Menotti (UFPR)
Daniel Fernando Pigatto (UTFPR)
Emerson Cabrera Paraiso (PUCPR)
Mauro Sérgio Pereira Fonseca (UTFPR)
Ricardo Luders (UTFPR)

Coordenação da Comissão Especial de Sistemas Multimídia e Web

Joel dos Santos (CEFET/RJ) – *Coordenador*
Leonardo Rocha (UFSJ) – *Vice-Coordenador*

Comitê Gestor

Adriano César Machado Pereira (UFMG)
Alan Guedes (UCL, UK)
Carlos de Salles Soares Neto (UFMA)
Carlos André G. Ferraz (UFPE)
Cezar Teixeira (UFSCar)
Débora Christina Muchaluat Saade (UFF)
Diego Roberto (UFSJ)
Eduardo Barrère (UFJF)
Fernando Mourão (Seek)
Jonice de Oliveira (UFRJ)
José Valdeni (UFRGS)
Jussara Almeida (UFMG)
Leyza Baldo Dorini (UTFPR)
Maria da Graça Campos Pimentel (USP)
Polyana Costa (PUC-Rio)
Roberto Willrich (UFSC)
Thiago Henrique Silva (UTFPR)
Valter Roesler (UFRGS)
Windson Viana de Carvalho (UFC)

Comitê de Programa de Minicursos e Tutoriais

Adriano César Machado Pereira (UFMG)
Adriano Machado Pereira (UFMG)
Alan Guedes (UCL, UK)
Carlos André G. Ferraz (UFPE)
Carlos de Salles Soares Neto (UFMA)
Celso Saibel Santos (UFES)
Cristiano Akamine (Mackenzie)
Débora C. Muchaluat Saade (UFF)
Diego Colombo Dias (UFSJ)
Fernando Mourao (SEEK)
Glauco Amorim (CEFET/RJ)
Joel dos Santos (CEFET/RJ)
Leonardo Rocha (UFSJ)
Ricardo Dutra da Silva (UTFPR)
Roberto Pereira (UFPR)
Rodrigo Minetto (UTFPR)
Windson Viana de Carvalho (UFC)

Sociedade Brasileira de Computação (SBC)

Presidência

Raimundo José de Araújo Macêdo (UFBA) – *Presidente*
André Carlos P. de L. F. de Carvalho (USP) – *Vice-Presidente*

Diretoria

Alírio Santos Sá (UFBA)
Carlos André Guimarães Ferraz (UFPE)
Carlos Eduardo Ferreira (USP)
Cristiano Maciel (UFMT)
Itana Maria de Souza Gimenes (UEM)
Jair Cavalcanti Leite (UFRN)
José Viterbo Filho (UFF)
Marcelo Duduchi Feitosa (CEETEPS)
Michelle Silva Wangham (UNIVALI)
Renata de Matos Galante (UFGRS)
Tanara Lauschner (UFAM)
Wagner Meira Júnior (UFMG)

Diretoria Extraordinária

Leila Ribeiro (UFRGS)

Contato

Av. Bento Gonçalves, 9500
Setor 4 - Prédio 43.412 - Sala 219
Bairro Agronomia
91.509-900 – Porto Alegre RS
CNPJ: 29.532.264/0001-78
http://www.sbc.org.br

Sumário

Capítulo 1

Identificação de Câmaras de Eco em Redes Sociais Através de Detecção de Comunidade em Redes Complexas: Ferramentas, Tendências e Desafios

Nicollas Rodrigues de Oliveira, Dianne Scherly Varela de Medeiros, Diogo Menezes Ferrazani Mattos

[1]LabGen/MídiaCom - PPGEET/TET/TCE/UFF
Universidade Federal Fluminense (UFF)
Niterói, Brasil

{nicollas_rodrigues,diannescherly,diogo_mattos}@id.uff.br

Resumo

A câmara de eco é um fenômeno relacionado à tendência de usuários de redes sociais interagirem com outros usuários em grupos homogêneos e com ideias e opiniões semelhantes. Como resultado, a câmara de eco prejudica o contraditório e incentiva o fenômeno do viés de confirmação, fomentando ambientes propícios ao discurso de ódio e à propagação de notícias falsas (fake news). Este minicurso apresenta os principais algoritmos para a caracterização estrutural e técnicas que auxiliam na detecção de câmaras de eco. O minicurso se concentra em abordagens de descoberta de comunidades sobre um grafo de topologia criado de acordo com a difusão de informações em redes sociais. Detalham-se também algoritmos de caracterização de redes complexas e os índices de desempenho dessas abordagens. Além disso, o minicurso desenvolve uma atividade prática de captura de dados em redes sociais e análise para identificação de câmaras de eco. Por fim, são discutidos os desafios e ferramentas que focam no estudo de câmaras de eco em redes sociais online.

1.1. Introdução

A facilidade de acesso pervasivo e ubíquo para a publicação e consumo de informações torna as plataformas de redes sociais um importante meio de interação pública.

Este capítulo foi realizado com recursos do CNPq, CAPES, RNP, FAPERJ, FAPESP (2018/23062-5) e Prefeitura de Niterói/FEC/UFF (Edital PDPA 2020).

Estudos recentes revelam que 71% dos brasileiros recorreram a plataformas como o *Facebook* e o *Twitter* para se informarem[1]. Apesar de hospedarem tanto conteúdos úteis para geração de conhecimento, quanto para o entretenimento, é latente a vocação das redes sociais em fomentar e repercutir conteúdos focado no discurso de ódio e em notícias falsas (*fake news*) [de Oliveira et al., 2020a]. Essa tendência de amplificar fenômenos indesejáveis é consequência direta da alteração do modelo de produção e consumo de informação. Tradicionalmente o processo de seleção de informação era mediado por jornalistas ou editores. Contudo, atualmente passou a ser exercido em redes sociais por todos e quaisquer usuários, com ou sem formação adequada para divulgação de notícias. Essa supressão ou ausência da mediação profissional especializada, contribui para degradação dos princípios de imparcialidade e legitimidade da informação, uma vez que usuários não treinados são mais suscetíveis ao fenômeno conhecido como "viés de confirmação". Tal fenômeno expressa a tendência dos usuários absorverem e compartilharem informações que se aderem ao seu sistema de crenças, mesmo que essas informações sejam inverídicas. Igualmente recorrente nas redes sociais, o efeito da "câmara de eco" é um fenômeno social relacionado à tendência dos usuários em interagirem e ingressarem em grupos homogêneos com ideias semelhantes às suas. Embutido no cerne desse fenômeno está o conceito de *unfriending* que consiste na exclusão intencional de usuários com opiniões contrárias às adotadas na câmara de eco [Cota et al., 2019, de Oliveira et al., 2021a, Zollo et al., 2017].

Implicitamente o fenômeno da câmara de eco engloba mecanismos que alienam os membros da comunidade formada, impedindo o acesso à fontes de informação epistêmicas externas. Assim, as câmaras de eco funcionam sistematicamente para formar padrões ideológicos disfuncionais que impedem os integrantes da câmara de eco de se envolverem em buscas informativas além da sua comunidade intelectual. Além de impedir o acesso à informação, os mecanismos envolvidos na formação das câmaras de eco desacreditam ativamente informações externas à câmara. Como consequência, há um desequilíbrio epistêmico que poderia implicar apenas em omissão de pontos de vista contrários, mas que, devido à formação da câmara de eco, implica em uma desconfiança preventiva ideológica dando origem ao aprisionamento dos indivíduos em câmaras de eco [Donkers e Ziegler, 2021].

Algumas redes sociais, como *Reddit*[2] e *Gab*[3], facilitam o processo de formação de câmaras de eco, uma vez que são naturalmente organizadas em comunidades relacionadas a um determinado tema. Recentemente, o *Whatsapp* divulgou o desenvolvimento da funcionalidade *Whatsapp Comunidades* capaz de agregar vários grupos em um mesmo espaço compartilhado[4]. Paralelamente, o *Twitter* lançou a *Roda do Twitter*[5], um recurso que permite que o usuário selecione a audiência do conteúdo a ser publicado, podendo encaminhá-lo somente a um grupo limitado e editável de amigos. Embora almejem facilitar a comunicação, tais recursos reacendem preocupações sobre os potenciais impactos nocivos da divulgação de conteúdos em ambientes virtuais segregados, ajudando a inten-

[1]Disponível em https://www.kaspersky.com.br/blog/pesquisa-infodemia-impactos-vida-digital/17467/.
[2]Disponível em https://www.reddit.com/.
[3]Disponível em https://gab.com/.
[4]Disponível em https://blog.whatsapp.com/sharing-our-vision-for-communities-on-whatsapp.
[5]Disponível em https://help.twitter.com/pt/using-twitter/twitter-circle.

sificar o viés de confirmação. Outro facilitador normalmente utilizado pelas plataformas de redes sociais é o algoritmo de recomendação, que fornece aos usuários mais do mesmo conteúdo consumido com base em seus comportamentos passados para moldar a preferência futura. Os usuários tendem a aceitar as recomendações e, adicionalmente, buscam ativamente mais informações sobre o conteúdo de interesse, devido ao viés de confirmação. Os algoritmos de recomendação também podem usar a popularidade de um conteúdo como indicativo de qualidade ou de preferência pessoal por um determinado conteúdo. No entanto, essa métrica de engajamento pode vir às custas da diversidade de opiniões [Sasahara et al., 2021]. Assim, o viés de confirmação e os algoritmos de recomendação criam uma espiral de auto-reforço. Como consequência, o ciclo de *feedback* entre algoritmos de recomendação e a psicologia humana eventualmente leva a uma câmara de eco que muda a visão de mundo dos usuários [Alatawi et al., 2021].

Um exemplo prático e preocupante das consequências da formação de câmaras de eco é o agravamento da polarização política nos últimos anos no mundo. Devido ao mecanismo de auto-reforço, as câmaras de eco ideológicas formadas alimentam ainda mais a polarização ao ampliar as lacunas de conhecimento entre grupos díspares. Recentemente, Fletcher *et al.* avaliaram o número de pessoas em câmaras de eco de notícias politicamente partidárias, abrangendo diferentes países europeus e os Estados Unidos. O estudo focou na contabilização de pessoas que usam apenas fontes de notícias *online* tendenciosas, sejam de esquerda ou de direita. Os resultados obtidos mostraram que no Reino Unido, cerca de 2% do público pesquisado integra uma câmara de eco inclinada para a esquerda, enquanto que cerca de 5% participam de uma câmara de eco inclinada para a direita. Tais proporções são semelhantes aos demais países analisados, com exceção dos EUA, onde estima-se que mais de 10% dos entrevistados confiam apenas em fontes de notícias partidárias [Fletcher et al., 2021].

O objetivo deste minicurso é apresentar os principais algoritmos, técnicas e métricas que auxiliam na caracterização estrutural e detecção de câmaras de eco (*echo chambers*) em redes sociais *online*. Diante das abordagens existentes de detecção de câmaras de eco, este minicurso foca na abordagem topológica. Essa abordagem explora a propagação do conteúdo entre os usuários e o padrão topológico de conexão entre eles para determinar a existência de uma comunidade. A Figura 1.1 ilustra o roteiro do minicurso, cujo foco está no processamento e detecção de câmaras de eco em redes sociais.

O restante do capítulo está estruturado conforme expresso a seguir. A Seção 1.2 descreve a metodologia utilizada. A Seção 1.3 define o fenômeno das câmaras de eco e o diferencia dos tópicos relacionados. As abordagens atuais de detecção de câmaras de eco são discutidas na Seção 1.4. As estratégias de criação de uma base de dados para identificação correta de câmaras de eco são apresentadas na Seção 1.5. A Seção 1.6 descreve como as câmaras de eco podem ser estruturadas, modeladas e caracterizadas utilizando conceitos de redes complexas. As Seções 1.7 e 1.8 explicam os processos para a transformação de textos e grafos em matrizes operáveis computacionalmente, respectivamente. A Seção 1.9 inclui uma descrição dos principais algoritmos de detecção de comunidade na literatura para identificar câmaras de eco. A Seção 1.10 elenca os desafios e as soluções focadas no combate de câmaras de eco. A Seção 1.11 apresenta uma atividade prática de identificação de câmaras de eco. A Seção 1.12 realiza as considerações finais do trabalho.

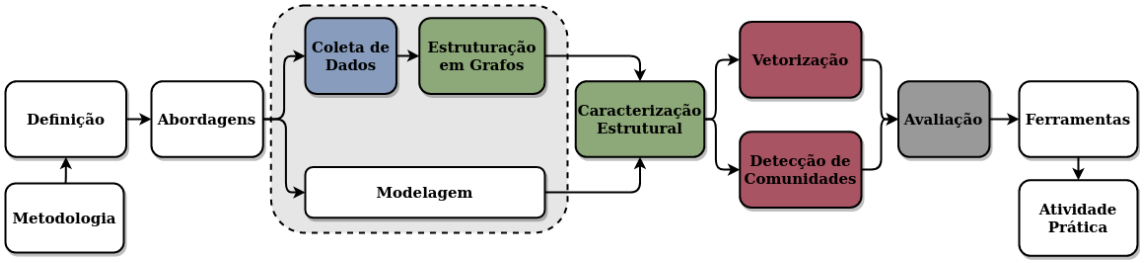


Figura 1.1. Fluxograma do roteiro do minicurso. Visando apresentar em detalhes o processamento e detecção de câmaras de eco sob uma perspectiva em grafos, o minicurso inicia relatando a metodologia adotada. Depois apresenta-se a definição de câmaras de eco seguida das diferentes abordagens de detecção da mesma. Posteriormente, debate-se que o estudo das câmaras de eco pode ser feito através de modelagem ou utilizando dados reais coletados e estruturados em grafos. Em seguida, descreve-se as principais métricas para caracterização das câmaras de eco, bem como os métodos de vetorização e algoritmos de detecção de comunidade. Além disso, são discutidas métricas de avaliação intrínsecas e extrínsecas. Vale ressaltar que a atividade prática aborda todas as etapas descritas neste roteiro, exceto aquelas em branco.

1.2. Metodologia

Neste minicurso, utiliza-se a metodologia PRISMA[6] (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) para realizar uma revisão sistemática da literatura a fim de identificar e recuperar estudos relevantes publicados em periódicos revisados por pares e anais de conferências.

A pesquisa bibliográfica é conduzida através de uma busca por palavras-chave claramente definidas na base de dados acadêmicos *online Dimensions*[7]. A escolha do *Dimensions* justifica-se pela sua capacidade de oferecer uma cobertura temporal e de fonte de publicação mais ampla do que o *Scopus* e *Web of Science* na maioria das áreas, sendo equiparável em cobertura ao *Google Scholar*. O levantamento inicial foca em buscas booleanas usando as palavras-chave: "*echo chamber*", em combinação com (AND) "*social media*" (OR) "*social network*". Ao buscar correspondências do conjunto de palavras-chaves escolhido nos títulos e resumos dos estudos, a ferramenta retorna 500 resultados. Todos os resultados são submetidos ao processo de avaliação de elegibilidade detalhado no diagrama de fluxo da metodologia PRISMA mostrado na Figura 1.2. Uma vez identificados, os 500 títulos são submetidos a uma etapa de triagem, onde são removidos eventuais duplicatas (n=28), selecionando os 472 estudos únicos. Ainda analisando os títulos e os resumos na etapa de triagem, uma segunda remoção é desempenhada com base nos seguintes critérios de exclusão: i) conteúdo não escrito na língua inglesa (n=18); ii) capítulos de livro, livros, monografias e artigos curtos (n=77); iii) publicação em revistas ou conferências de áreas outras áreas do conhecimento, tais como ciência política, jornalismo ou psicologia (n=197). Fundamentando-se nesses critérios de exclusão, apenas 180 artigos prosseguiram para uma verificação mais detalhada, na qual o texto completo de cada artigo é avaliado. Nesta etapa, são igualmente excluídos artigos sem acesso aberto (n=52) e artigos cuja proposta não é diretamente relacionada à câmaras de eco (n=53).

[6]Disponível em https://www.prisma-statement.org/.

[7]Disponível em https://www.dimensions.ai/.

Dessa verificação em profundidade restam 75 estudos aptos a serem analisados qualitativamente. Contudo, por questões de espaço, este minicurso seleciona e analisa 30 artigos.

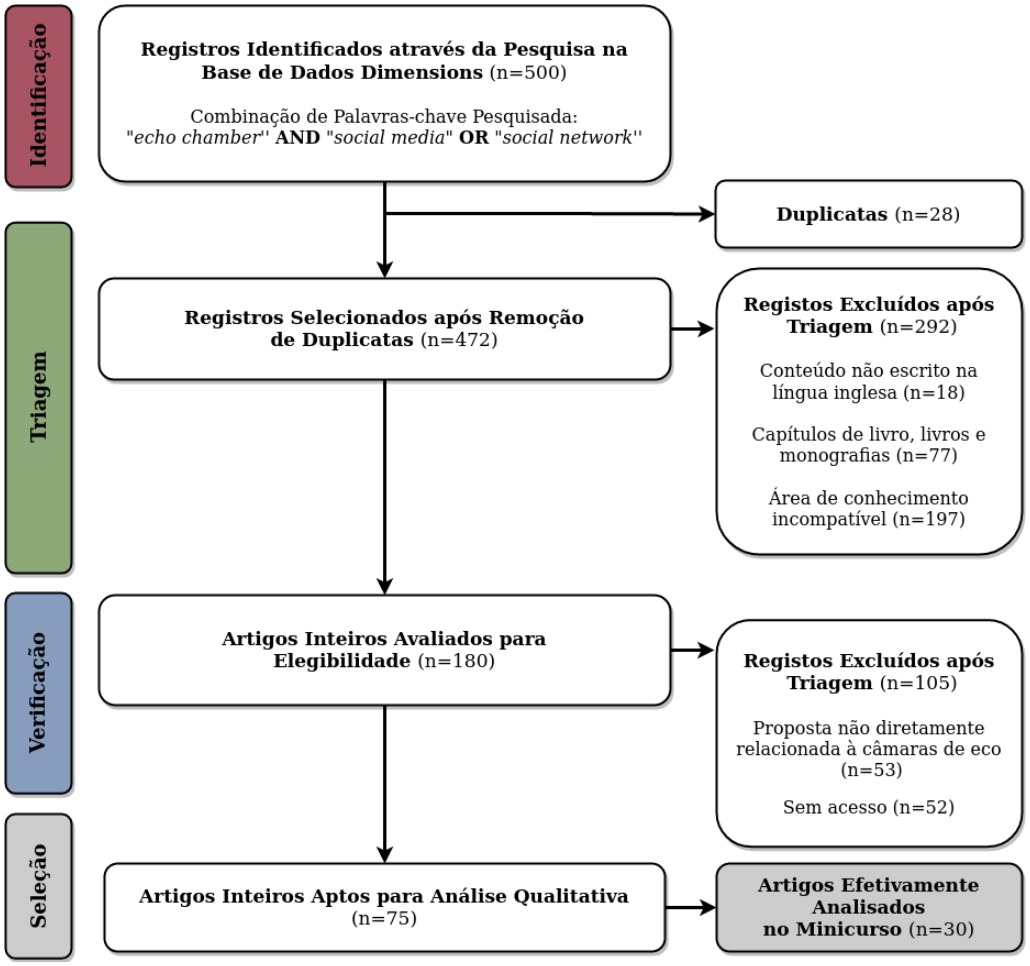


Figura 1.2. Fluxograma da metodologia Prisma aplicada ao tema abordado no minicurso: câmaras de eco. Para formar um conjunto inicial de artigos a respeito do tema, realiza-se uma pesquisa bibliográfica na base de dados *Dimensions* utilizando palavras-chaves específicas. Seguindo vários critérios de exclusão, seleciona-se apenas os artigos científicos publicados em inglês e estritamente alinhados com o tema e com metodologias computacionais.

1.3. Caracterização das Câmaras de Eco

As câmaras de eco não possuem uma definição única e consensual na literatura. Apesar da multiplicidade de definições, em um sentido mais amplo, *as **câmaras de eco** são definidas como uma rede ativa de usuários, na qual ideias semelhantes são amplificadas ou reforçadas por meio de um processo de compartilhamento repetitivo de uma mesma ideia, blindando-a de refutação* [Morini et al., 2021]. Diante dessa definição, caracteriza-se uma câmara de eco em relação a três aspectos: i) a composição estrutural, ii) o conteúdo divulgado e iii) o comportamento de seus integrantes perante refutação. Nas redes sociais *online*, as câmaras de eco organizam-se na forma de uma rede de comunicação altamente conectada, composta por diferentes tipos de usuários, porém com posicionamentos ideológicos semelhantes. Dentro das câmaras de eco, há uma preva-

lência pelo compartilhamento de conteúdos extremamente unilaterais e muito aderentes às posturas e opiniões dos usuários que a integram. Essa ressonância de informação no interior de câmaras de eco é igualmente acompanhada por uma tática ativa de descredibilização e exclusão de quaisquer fontes ou opiniões externas ou divergentes. Na prática, quando expostos a fatos que desafiam suas crenças, os membros de câmaras de eco rapidamente os rejeitam já que não fortalecem seus posicionamentos originais. Tal atitude reativa a contra-evidências é descrita pelo efeito *backfire* [Alatawi et al., 2021], que resulta em membros acreditarem em informação ilegítima, mesmo após a apresentação de evidências de inverdades.

A definição de câmara de eco confunde-se comumente com a de outro fenômeno relacionado, a **bolha de filtro** (*filter bubble*). Embora a bolha de filtro também restrinja o acesso à informação de seus integrantes, o isolamento observado nas bolhas de filtro é derivado do uso de um conjunto extremamente restrito de recomendações. Normalmente, essas sugestões são automaticamente realizadas pelos chamados sistemas de recomendação, empregados em redes sociais *online*. Esses sistemas de recomendação consistem em algoritmos que inferem os interesses ou preferências do próprio usuário por meio da análise de dados subjacentes ao uso da rede, tais como comportamento de cliques, histórico de compras ou registro de pesquisas [Ge et al., 2020]. Como a superabundância de informações seria esmagadora para qualquer usuário, os sistemas de recomendação têm o papel fundamental de personalizar o fluxo de informações, selecionando e exibindo somente o conteúdo mais relevante para cada usuário. As redes sociais exigem uma quantidade adequada de diversidade para permitir a soberania informacional de seus usuários. No entanto, a imposição de filtros personalizados nas informações pode exacerbar a fragmentação, criando ciclos de *feedback* degenerados, nos quais a amplitude da informação é cada vez mais reduzida ao longo do tempo [Donkers e Ziegler, 2021].

Comparativamente, a bolha de filtro apresenta uma estrutura relativamente mais frágil que a câmara de eco, uma vez que são permissivas a fontes externas relevantes. Ao contrário da descredibilização adotada nas câmaras de eco, as fontes externas são normalmente ignoradas por motivos involuntários do usuário. Diferentemente das câmaras de eco, a exposição dos usuários em bolhas de filtro à informações ou argumentos relevantes, não experienciados antes, permite libertá-los de suas bolhas [Nguyen, 2020]. Embora menos potencialmente danosas, a superpersonalização intencional de uma bolha de filtro pode eventualmente confinar usuários em câmaras de eco. Outro termo relacionado, o ***gatekeeping***, traduz-se como uma prática de filtragem de informação tradicionalmente executada por fontes midiáticas. Quando essa prática é executada por usuários, esses passam a ser intitulados como *gatekeepers*. Os *gatekeepers* consomem conteúdo de múltiplas visões ideológicas, porém reproduzem ou disseminam conteúdos relacionados a uma única visão [Garimella et al., 2018a].

A organização dos usuários em grupos homogêneos está intimamente atrelada ao conceito de **homofilia**, ou seja, a tendência dos indivíduos de se associarem a outros semelhantes. Essa semelhança ou similaridade social pode ser classificada em: i) homofilia de *status*, relacionada ao agrupamento de pessoas com base em suas características semelhantes atribuídas (sexo, raça ou etnia) ou adquiridas (educação ou religião); e ii) homofilia de valor, que envolve agrupar pessoas semelhantes com base em seus valores, atitudes ou crenças. Dependendo da ideologia atrelada, a câmara de eco pode ser for-

mada devido à homofilia de *status*, homofilia de valor ou ambos. As plataformas de redes sociais afrouxaram as fontes básicas de homofilia, como a geografia, permitindo que os usuários vinculem relacionamentos homófilos em outras dimensões, como raça, etnia, sexo, gênero e religião [Colleoni et al., 2014, Alatawi et al., 2021].

As câmaras de eco podem tornar os usuários de mídia social mais vulneráveis a esse tipo de manipulação. A estrutura da câmara de eco pode contribuir para a disseminação de **notícias falsas** de diversas formas. Em câmaras de eco, as pessoas são repetidamente expostas a informações homogêneas. Além disso, a seleção de informações consistentes com as crenças e a prevenção de informações contrárias ou contrastantes às crenças reforçam a confiança nas opiniões minoritárias, como notícias fabricadas, mesmo na presença de evidência contrária preponderante. A estrutura da câmara de eco também pode induzir a uma convergência rápida e prematura para soluções sub-ótimas de problemas complexos. Ademais, o limiar para perceber um conteúdo como novo pode ser menor dentro das câmaras de eco em virtude da reduzida diversidade de pontos de vista aos quais as pessoas estão expostas. As tentativas de captura de atenção são desempenhadas pelo uso de conteúdo inflamatório, polêmico, ou emocional [Törnberg, 2018].

Além das notícias falsas, as câmaras de eco criam uma estrutura que potencializa a crença de que alguma organização secreta, mas influente, se reúne em acordo secreto a fim de alcançar um objetivo malévolo. Na prática, as **teorias conspiratórias** são tentativas de explicar as causas finais de eventos e circunstâncias sociais e políticas significativas. Seus adeptos usam as redes sociais para se encontrarem, disseminarem conteúdos conspiratórios e compartilharem pontos de vista marginais. As teorias da conspiração expressam e amplificam ansiedades e medos de perder o controle da ordem religiosa, política ou social. Na literatura científica, diversos trabalhos debruçam-se sobre o estudo das câmaras de eco considerando a disseminação de teorias conspiratórias em seu interior [Del Vicario et al., 2016, Bessi, 2016]. Com o objetivo de esclarecer e evitar o intercambiamento equivocado, a Tabela 1.1 condensa as definições dos principais termos e conceitos associados às câmaras de eco.

1.3.1. Trabalhos Relacionados

Na literatura científica, o estudo de câmaras de eco é frequentemente associado a outros assuntos relacionados, dentre eles: a controvérsia ou polarização ideológica, a disseminação de conteúdo falso, a modelagem de opinião, a homofilia e sistemas de recomendação. Dentre esses trabalhos, Cossard *et al.* analisam no *Twitter* o debate sobre vacinação tendo como premissa a existência de câmaras de eco [Cossard et al., 2020]. Ao caracterizar e distinguir usuários em grupos de acordo com seus posicionamentos defensivo ou cético sobre vacinas, as características relacionais e textuais aplicadas a algoritmos de classificação. Resultados qualitativos apontam que céticos e os defensores da vacinação residem em suas próprias câmaras de eco e têm preferências por fontes de informação excludentes entre si. Visando promover a redução da controvérsia em redes sociais, Garimella *et al.* investigam técnicas algorítmicas para interligar usuários de câmaras de eco isoladas [Garimella et al., 2017]. Ao empregar uma métrica de quantificação de controvérsias, o modelo proposto busca pelas chamadas pontes, conexões entre usuários que minimizem essa métrica. Partindo da hipótese de que usuários localizados na borda de uma câmara de eco estão menos fechadas à diferentes opiniões, espera-se que conteúdos

Tabela 1.1. Definições dos termos e conceitos associados à câmaras de eco.

Termos	Definição
Câmaras de Eco	Estruturas sociais que excluem sistematicamente as fontes de informação não necessariamente por omissão, mas por ação deliberada.
Bolha de Filtro	Ambiente social superpersonalizado derivado da filtragem excessiva de informação, praticada automaticamente por sistemas de recomendação.
Sistemas de Recomendação	Algoritmos que inferem os interesses ou preferências do próprio usuário por meio da análise de dados subjacentes ao uso da rede, exibindo em um conteúdo personalizado.
Gatekeeping	Prática de filtragem de informação tradicionalmente executada por fontes de mídia.
Homofilia	Conceito psicológico relacionado à tendência dos indivíduos de se associarem a outros com características ou valores semelhantes.
Viés de Confirmação	Conceito psicológico relacionado à tendência dos indivíduos procurarem e privilegiarem o recebimento de informações que refiram seus pontos de vista existentes.
Notícias Falsas	Notícias comprovadamente inverídicas, criadas e disseminadas intencionalmente para confundir ou descredibilizar algo ou alguém.
Teorias de Conspiração	Crença na existência de que alguma organização secreta, mas influente, se reúne em acordo secreto a fim de alcançar um objetivo malévolo.

enviados de fora da câmara de eco possam ser recebidos e eventualmente endossados pelo usuário que o recebeu.

Ge *et al.* investigam a formação e os efeitos de câmaras de eco em sistemas de recomendação de plataformas *e-commerce* [Ge et al., 2020]. Empregando uma base de dados real formada pelo histórico de acesso *web* dos usuários, os autores distinguiram esses usuários em grupos com base na frequência que estes aceitam, ou ignoram, produtos recomendados. A análise temporal dos resultados sugere que as câmaras de eco apresentam um impacto maior sobre o comportamento de clique do usuário, enquanto que têm sua influência mitigada sobre comportamentos de compra. Tentando prevenir a superpersonalização dos sistemas de recomendação, Dash *et al.* apresentam um *framework* baseado em redes para auditar de sistemas de recomendação [Dash et al., 2019]. Em particular, os autores quantificam duas propriedades principais, a diversidade das recomendações fornecidas pelos sistemas de recomendação e o grau de segregação e polarização de informações fornecidas aos usuários. A proposta é especialmente adequada para uso por auditores terceirizados, uma vez que independe da disponibilidade de informações sobre a interação entre usuário e item. Com um objetivo mais específico, Cinus *et al.* visam avaliar a contribuição dos sistemas de recomendação de pessoas no aumento ou diminuição das câmaras de eco [Cinus et al., 2022]. Diferentemente dos tradicionais, os sistemas de recomendação de pessoas dedicam-se à geração de sugestões de amizade contendo pessoas

que compartilham os mesmos interesses, ou o mesmo grupo de amigos, que o usuário. Ao combinar um modelo de dinâmica de opinião com um algoritmo de recomendação de pessoas, a proposta é capaz de simular o comportamento de indivíduos mudando de opinião como consequência de suas interações com sua vizinhança. Resultados mostraram que, caso haja uma homofilia inicial considerável na rede, as recomendações de pessoas podem efetivamente contribuir para a inclusão de novos usuários em câmaras de eco.

Outros trabalhos visam traçar relações de causa e efeito entre câmaras de eco e disseminação de conteúdo falso. Com esta finalidade, Törnberg modela a disseminação de notícias falsas nas mídias sociais atribuindo a cada usuário um limite que descreve o quão difícil é convencê-lo sobre uma determinada narrativa [Törnberg, 2018]. Caso uma fração suficientemente grande de seus vizinhos espalhe um mesmo conteúdo, o usuário é convencido e replicará o conteúdo. O modelo revela que as notícias originadas no interior de câmaras de eco têm uma maior disseminação do que quando comparado a uma rede sem grupos definidos. Além disso, mostra-se que a simples reunião de usuários com visões homogêneas pode ser suficiente para aumentar a prevalência de desinformação, tendo em vista que a viralidade é diretamente proporcional à homofilia da rede. De forma similar, Bessi *et al.* conduzem uma análise quantitativa minuciosa para investigar a consumo e propagação de diferentes assuntos de teorias da conspiração dentro de câmaras de eco no *Facebook* [Bessi et al., 2015]. Ao coletar e analisar as curtidas, compartilhamentos e comentários dos usuários, revela-se que os diferentes assuntos conspiratórios são consumidos de maneira semelhante pelos usuários de câmaras de eco. A única divergência está no tempo de atividade, ou seja, o intervalo temporal entre o primeiro e o último comentário de cada usuário. Em especial, usuários polarizados em assuntos relacionados à geopolítica são mais ativos nos comentários.

Em contrapartida, a modelagem psico-linguística explora as cinco dimensões básicas que compõem a personalidade humana para identificar o comportamento dos indivíduos envolvidos em narrativas de apoio dentro de câmaras de eco. As cinco dimensões básicas são: extroversão, estabilidade emocional, amabilidade, consciência e abertura. Uma das abordagens existentes, intitulada reconhecimento de personalidade não supervisionada, executa uma série de correlações estatisticamente significativas entre os traços de personalidade e as características linguísticas extraídas por meio de técnicas de Processamento de Linguagem Natural (PLN) [Bessi, 2016].

1.4. Abordagens de Detecção de Câmaras de Eco

Atualmente, as abordagens de detecção de câmaras de eco distinguem-se em duas grandes classes de abordagens, a baseada em ideologia e a baseada em topologia de rede. A **abordagem baseada na ideologia** infere a existência de um ambiente virtual polarizado ao analisar semanticamente a inclinação do conteúdo compartilhado ou consumido por um usuário [Alatawi et al., 2021]. Internamente, essa análise semântica visa medir a similaridade, ou distância, entre representações vetoriais do texto compartilhado, seja esse derivado de comentários, postagens, notícias ou mensagens. Tais representações vetoriais podem ser obtidas por meio de modelos simples como Frequência do Termo – Inverso da Frequência nos Documentos (*Term Frequency–Inverse Document Frequency*, TF-IDF), ou por meio de algoritmos de incorporação de palavras (*word embeddings*), que empregam redes neurais treinadas com grandes volumes de dados textuais. Algoritmos

como *GloVe* e *Word2Vec* são capazes de embutir a semântica das palavras em vetores de baixa dimensão, densos e de tamanho fixo, garantindo que sinônimos ou palavras minimamente relacionadas sejam mapeados em vetores similares [de Oliveira et al., 2021b]. Essa representação vetorial individualizada das palavras é especialmente empregada na detecção de notícias falsas, um conteúdo frequentemente divulgado em câmaras de eco.

A detecção de câmaras de eco segundo a **abordagem baseada em topologia de rede** explora os padrões de propagação de conteúdo entre os usuários. Normalmente, a abordagem topológica inicia com a extração de metadados de usuários participantes de um debate controverso nas redes sociais. Em posse dos metadados, é possível inferir os posicionamentos de cada usuário a respeito do tópico selecionado, bem como construir estruturas em grafos que representem as interações digitais entre eles. Dentre as vantagens dessa estruturação em redes de usuários estão: i) a possibilidade de revelar padrões estruturais característicos de câmaras de eco por meio de métricas de redes complexas; e ii) a identificação de grupos de usuários homogêneos a partir da aplicação de algoritmos de detecção de comunidades [Alatawi et al., 2021].

Morini *et al.* introduzem a visão de classificação das câmaras de eco de acordo com a escala da detecção. Nessa visão, as chamadas câmaras de eco de **microescala** são delimitadas avaliando o comportamento online de usuários singularmente, fato que acarreta a perda de sua dimensão agregada. Alternativamente, as câmaras de eco de **macroescala** são delimitadas considerando a rede de interação dos usuários em um nível mais amplo, ou seja, analisando o grafo como um todo. Contudo, este tipo de câmara de eco pode ignorar eventuais diferenças dentro de determinadas áreas da rede. Uma terceira classificação expressa que as câmaras de eco de **mesoescala** podem ser identificadas como apenas um subconjunto de nós na rede geral. Isso implica que, na rede geral de debates, é possível encontrar diversas câmaras de eco com a mesma inclinação ideológica [Morini et al., 2021]. A Tabela 1.2 apresenta vários trabalhos relacionados ao estudo de câmaras de eco, distinguindo-os segundo a abordagem adotada e o tipo da câmara de eco detectada.

1.5. Construção da Base de Dados

A construção de uma base de dados potencialmente relacionada às câmaras de eco inicia-se com a seleção de um tópico controverso. Independente do domínio de conhecimento, a discussão sobre questões duais propiciam a polarização de opiniões e podem transformar atitudes divergentes em extremos ideológicos. Questões controversas são igualmente debatidas tanto no âmbito *offline* quanto no *online*. Contudo as plataformas de redes sociais *online* são provavelmente o espaço aberto mais utilizado para discussões. Essas plataformas possuem estrutura e funcionalidades que facilitam a identificação de debates *online* sobre uma ampla gama de questões diferentes. Plataformas como o *Twitter*, *Instagram* e *Facebook* permitem que os usuários incluam em suas postagens as*hashtags*, que são palavras, ou frases sem espaçamento, prefixadas com o caractere cardinal "#". Embora tenham sido originalmente idealizadas para indexar as postagens dos usuários, as *hashtags* atualmente comportam-se como um termômetro de eventos sociopolíticos, comerciais e culturais. Visto que as *hashtags*, assim como qualquer outra manifestação textual, expressam os interesses, opiniões e crenças dos seus autores, elas podem ser usadas como ponto de partida para encontrar grupos de usuários com posicionamentos se-

Tabela 1.2. Trabalhos relacionados à modelagem, caracterização, quantificação ou detecção de câmaras de eco.

	Abordagem		Tipo de Câmara de Eco		
	Topológica	**Ideológica**	**Microescala**	**Mesoescala**	**Macroescala**
[Bakshy et al., 2015]	✗	✓	✓	✗	✗
[Zollo et al., 2017]	✗	✓	✗	✗	✓
[Morini et al., 2021]	✓	✓	✗	✓	✗
[Conover et al., 2011]	✓	✓	✗	✗	✓
[Barberá et al., 2015]	✓	✓	✗	✗	✓
[Cossard et al., 2020]	✓	✓	✗	✗	✓
[Villa et al., 2021]	✓	✓	✗	✗	✓
[Williams et al., 2015]	✓	✓	✗	✗	✓
[Sasahara et al., 2021]	✓	✗	✗	✗	✓
[Bessi, 2016]	✗	✓	✗	✗	✓
[Garimella et al., 2018a]	✓	✓	✗	✗	✓
[Morales et al., 2021]	✓	✓	✗	✗	✓
[Cinelli et al., 2021]	✓	✓	✗	✗	✓
[Zannettou et al., 2018]	✗	✓	✗	✗	✓
[Cota et al., 2019]	✓	✗	✗	✗	✓
[Baumann et al., 2020]	✓	✗	✗	✗	✓
[Gillani et al., 2018]	✓	✗	✗	✗	✓

melhantes sobre um determinado assunto [de Oliveira et al., 2021a]. De forma diferente, plataformas como *Reddit* e *Gab* são organizadas como fóruns virtuais que restringem o debate por temas de interesse, tirando o foco das *hashtags*. No *Reddit* tais comunidades de interesse são chamadas *subreddits*. Para coletar os dados oriundos dessas redes de forma automática, algumas plataformas disponibilizam APIs, como o *Twitter*[8]. Além de utilizar a API própria da plataforma, é possível utilizar códigos terceirizados desenvolvidos no formato de APIs, como no caso do *Reddit* e do *Gab*[9]. A Tabela 1.3 traz uma compilação de bases de dados presentes na literatura sobre câmaras de eco ou qualquer tópico relacionado, como polarização política e bolhas de filtro.

1.6. Estruturação e Modelagem de Câmaras de Eco

A Análise de Redes Sociais (*Social Network Analysis* - SNA) compreende a investigação de relações ou fenômenos sociais por meio do uso de grafos, que são estruturas matemáticas compostas por vértices e arestas. Um grafo $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ possui $|\mathscr{V}|$ vértices (ou nós) interligados por $|\mathscr{E}|$ arestas. Os grafos são orientados quando a relação entre dois nós é unidirecional e, portanto, para que $v_i \in \mathscr{V}$ esteja conectado a $v_j \in \mathscr{V}$ e v_j esteja conectado a v_i devem existir duas arestas $\{\varepsilon_{i,j}, \varepsilon_{j,i}\} \in \mathscr{E}$. Se o grafo é não orientado, a relação entre nós é bidirecional, e apenas uma aresta $\varepsilon_{i,j} \in \mathscr{E}$ precisa existir entre v_i e v_j para que estejam conectados entre si. As arestas, sejam bidirecionais ou unidirecionais, podem possuir pesos não unitários $w_{i,j} \in \mathscr{W}$, transformando os grafos em grafos ponderados $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{W})$. Os pesos representam a intensidade da ligação entre os pares de nós adjacentes.

[8]Disponível em https://developer.twitter.com/en/docs/twitter-api.

[9]Disponível em https://github.com/pushshift/api.

Tabela 1.3. Base de dados

	Plataforma	Tópico	#P	#U	Período
[Bessi, 2016]	Facebook	Ciência e Teorias Conspiratórias	3M	30k	2010-2014
[Del Vicario et al., 2016]	Facebook	Ciência e Teorias Conspiratórias	271k	ND	2010-2014
[Garimella e Weber, 2017]	Twitter	Orientação Política	ND	140M	2009-2016
		Orientação Política	2 bi	679k	2007-2016
[Garimella et al., 2018a]	Twitter	Controle de Armas	19M	7.5k	2016
		Política de Saúde (*obamacare*)	39M	8.7k	2015
		Legalização do Aborto	34M	3.9k	2016
[Cossard et al., 2020]	Twitter	Vacinação	818k	102k	2018-2019
[Cota et al., 2019]	Twitter	Orientação Política	12M	285k	2016
[Gillani et al., 2018]	Twitter	Orientação Política	ND	1.1M	2016
[Barberá et al., 2015]	Twitter	Politica, Esporte e Entretenimento	150M	3.8k	2012-2014
[Morini et al., 2021]	Reddit	Orientação Política	431k	72k	2017-2019
		Controle de Armas	180k	65k	
		Discriminação de Minorias	223k	52k	
[Cinelli et al., 2021]	Reddit	Orientação Política	353k	240k	2017
	Reddit	Personalidade *(the donald)*	1.2M	138k	
	Reddit	Notícias	723k	179k	
	Gab	ND	13M	165k	
[Zannettou et al., 2018]	Gab	ND	22M	336k	2016-2018

#P representa o número de postagens.
#U representa o número total de nós.
ND representa informações não disponibilizadas pelos autores.

As redes sociais *online* são representadas e modeladas matematicamente como redes complexas, *i.e.*, grafos detentores de um grande número de nós e interligados por meio de uma topologia complexa, podendo esses grafos serem orientados ou não, dependendo do tipo de relação entre nós. Por exemplo, em uma rede social como *Twitter* ou *Instagram* os grafos que representam a conexão entre pessoas é orientado. A pessoa v_i pode seguir a pessoa v_j sem que a pessoa v_j siga a pessoa v_i, existindo, assim, apenas a

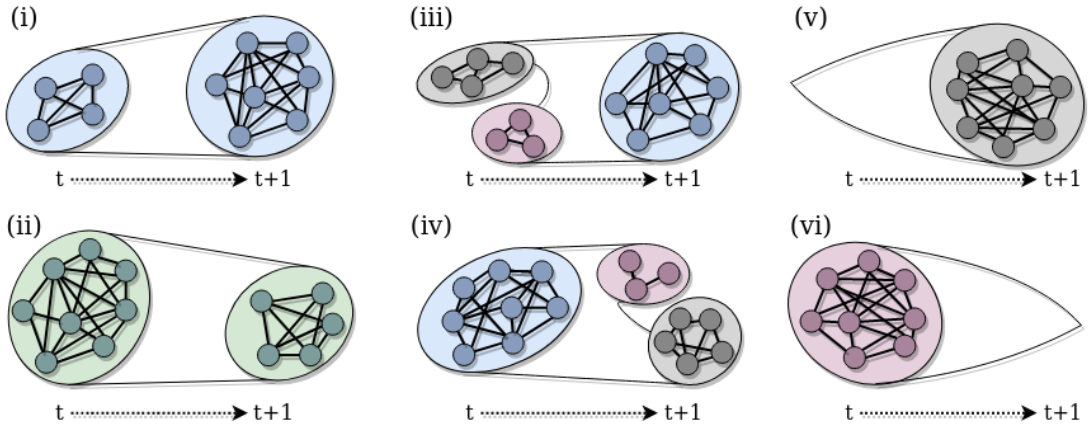


Figura 1.3. Principais eventos na evolução de comunidades ao longo do tempo: (i) crescimento, (ii) contração, (iii) mesclagem, (iv) divisão, (v) nascimento e (vi) morte. Adaptado de [Barabási, 2016].

aresta $\varepsilon_{i,j}$. Isso implica que v_i recebe o conteúdo de v_j, mas o inverso não é verdade. Em uma rede social como *Facebook*, para que duas pessoas estejam conectadas, elas precisam ser amigas uma da outra, de forma que há uma relação bidirecional entre todos os nós que fazem parte dessa rede. Assim, o grafo da rede para essa representação de conexão entre pessoas é não orientado e todas as arestas representam uma relação bidirecional em que v_i recebe conteúdo de v_j e vice-versa.

O problema abstrato de identificar câmaras de eco em redes sociais é tratado na abordagem topológica como um problema de identificar comunidades em um grafo. Na teoria de redes complexas, define-se uma comunidade como um subconjunto de nós densamente conectados entre si e esparsamente conectado aos demais nós [Yang et al., 2016, Terren e Borge-Bravo, 2021]. Em uma rede estática, a topologia não se altera ao longo do tempo, facilitando a aplicação direta de algoritmos clássicos de detecção de comunidades. Porém, essa estruturação estática não é suficientemente capaz de retratar a natureza evolutiva de sistemas reais, como a criação de novos vínculos com outras pessoas e a extinção de antigos vínculos, o que representaria, por exemplo, novas amizades e o término de antigas amizades. Assim, devido à dinamicidade das redes sociais, a topologia da rede varia ao longo do tempo e a rede estática não é a melhor representação. Nesse sentido, é válido utilizar representação de redes temporais para modelar a rede social em observação. As redes temporais têm uma topologia variável no tempo, em que nós e arestas podem estar ativos ou não, dependendo do instante de observação. Ao lidar com redes temporais, seis principais eventos podem ocorrer com as comunidades entre os tempos de observação t e $t+1$, conforme retratado na Figura 1.6. O (i) Crescimento ocorre quando há um aumento do tamanho de uma comunidade entre os tempos de observação. A (ii) Contração ocorre se houver diminuição do tamanho da comunidade. Duas ou mais comunidades podem ser unidas, incorrendo o evento de (iii) Mesclagem. De forma contrária, uma comunidade pode ser separada em duas ou mais comunidades pelo evento da (iv) Divisão. Por fim, podem ocorrer o (v) Nascimento, quando uma comunidade aparece pela primeira vez, e a (vi) Morte quando uma comunidade desaparece.

Existem três modelos principais para representação de redes temporais:

- **Sequência de Contato**, redes cujas arestas existem por um tempo desprezível e são representadas como um conjunto de sequência de contato (i, j, t) em que i e j são nós, t é o carimbo de tempo do contato;

- **Grafo de Intervalo**, redes que registram a duração dos períodos de atividade da uma aresta e através um conjunto de intervalos $T_e = (t_1, t'_1)...(t_n, t'_n)$;

- **Séries de Instantâneos**, rede segmentada em um conjunto de janelas de tempo sequenciais (e.g. *snapshot*) em que contatos em uma janela de tempo são agregados em arestas. A evolução da rede temporal pode ser estudada através desses *snapshots*.

Embora as redes temporais de **sequência de contato** e **grafo de intervalo** preservem as informações temporais, esses modelos têm uma análise mais complexa e dependem da aplicação de novas metodologias e algoritmos. A rede temporal representada por uma **série de instantâneos** reduz a complexidade da análise, permitindo o uso de métodos de pesquisa de redes estáticas de forma independente em cada *snapshot*. Contudo, a principal limitação do modelo instantâneo é a indeterminação quanto ao tamanho ótimo do grafo para quaisquer análises. Na prática, a escolha do tamanho adequado depende de uma profunda compreensão da rede em questão, sendo vital para evitar a particularização da análise e a obtenção de resultados dissonantes. *Snapshots* muito curtos podem não conter arestas suficientes agregadas em cada instantâneo, resultando em informações incompletas. Por outro lado, o uso de *snapshots* muito longos pode mascarar detalhes da evolução do grafo [Rossetti e Cazabet, 2018].

Mediante a coleta de dados, muitos autores costumam estruturar as câmaras de eco aproveitando-se dos dados nativos extraídos das plataformas de redes sociais. Com o *Twitter*, por exemplo, pode-se empregar metadados descorrelacionados como *retweets*, menções e seguidores para construir grafos relacionais. Esses grafos representam redes semânticas em que os nós descrevem um determinado padrão e as arestas descrevem a relação entre os nós, ou seja, a relação entre os padrões [Meyer-Baese e Schmid, 2014]. Assim, as interações usuário-usuário, como seguir, repostar, bater papo e comentar entre si, podem ser utilizadas para desvendar a difusão de informações e a relação entre os usuários das redes sociais *online*. Uma rede de *retweets* pode ser reproduzida através de um grafo relacional direcionado ponderado em que os nós representam o conjunto de usuários distintos e o peso de uma aresta do nó v_i ao nó v_j representa o número de vezes que o usuário v_i reposta um *tweet* do usuário v_j. Essa mesma lógica pode ser adotada com plataformas que também disponibilizam o número de curtidas (*likes*) nas postagens, como o *Facebook*. Similarmente, uma rede de menção pode ser retratada por um grafo relacional direcionado ponderado na qual os nós representam os usuários e as arestas representam as menções, ou seja, a ação de incluir um nome de usuário em um *tweet*. O peso de uma aresta do nó v_i ao nó v_j representa o número de vezes que o usuário v_i menciona o usuário v_j. Finalmente, a rede de seguidores pode ser construída por meio de um grafo relacional direcional na qual os nós representam usuários e uma aresta do nó v_i ao nó v_i representa o usuário v_i seguindo o usuário v_v.

1.6.1. Métricas de Redes Complexas em Comunidades

O mapeamento das redes sociais em grafos permite interpretar cada câmara de eco como uma comunidade, podendo assim revelar características estruturais latentes dessas câmaras a partir de métricas de redes complexas. Para o cálculo da maioria das métricas de redes complexas é necessário conhecer os conceitos de caminho, passeio, passeio aleatório e clique. Um caminho é definido como uma sequência ordenada de arestas que unem nós adjacentes, sem repetição dos nós. O tamanho desse caminho é definido pelo número de arestas atravessadas. O caminho mais curto, $\delta^*_{i,j}$, entre um par de nós é aquele que atravessa o menor número de arestas possível. De forma semelhante, um passeio também é uma sequência ordenada de arestas que unem nós adjacentes. Contudo, em um passeio, não há restrição quanto à repetição de nós e de arestas na sequência ordenada. Já um passeio aleatório é definido como um processo aleatório que parte de um nó inicial com destino a um nó final. Em cada nó a partir do nó inicial, há uma determinada probabilidade de seguir para um dos nós adjacentes ao nó atual. Ao alcançar o nó final, existe uma sequência ordenada de arestas que atravessa o grafo partindo do nó inicial até o nó final. A essa sequência gerada pelo processo aleatório dá-se o nome de passeio aleatório. Assim como no passeio tradicional, pode haver repetição de vértices e arestas. Por fim, um clique é um subconjunto de nós de um grafo não direcionado em que quaisquer dois nós distintos desse subconjunto são sempre adjacentes, isto é, o subconjunto de nós é totalmente conectado, formando uma malha completa.

Métricas comumente utilizadas na análise de comunidades são a densidade, o diâmetro, a assortatividade, e as diversas variações de centralidades [de Oliveira et al., 2021a]. A densidade de um grafo, $D(\mathscr{G})$, indica quão denso um grafo é em termos de conectividade de arestas e é medida em função do número total de arestas $|\mathscr{E}|$ e nós $|\mathscr{V}|$ pertencentes ao grafo, conforme Equação 2. O diâmetro é expresso pela máxima excentricidade do grafo, ou seja, o maior dentre todos os caminhos mais curtos existentes entre todos os pares de nós v_i e v_j, cujo comprimento é dado por $\delta^*(v_i, v_j)$, em número de saltos. Matematicamente, a excentricidade $e(\mathscr{G})$ é dada pela Equação 3. Assim, o diâmetro, $d(\mathscr{G})$, de um grafo, dado pela Equação 2, pode assumir valores inteiros no intervalo de $[1, \infty[$.

$$D(\mathscr{G}) = \frac{2|\mathscr{E}|}{|\mathscr{V}| \cdot (|\mathscr{V}| - 1)} \quad (1) \qquad d(\mathscr{G}) = \max_{v_i \in \mathscr{V}} e(v_i) \quad (2)$$

$$e(v_i) = \max_{v_j \in \mathscr{V}} \delta^*(v_i, v_j) \quad (3)$$

Quando transportado para o âmbito de redes complexas, o conceito de homofilia é representado pela métrica de assortatividade. Essa métrica, definida entre $[-1, 1]$, expressa a tendência de nós se conectarem a outros nós com valores semelhantes de uma determinada característica. Ao considerar o grau como característica a ser avaliada, por exemplo, valores positivos de assortatividade indicam uma correlação entre nós de grau semelhante, enquanto valores negativos indicam relações entre nós de grau diferente. Valores nulos traduzem a completa conexão entre todos os nós em um grafo. Casos extremos, positivos ou negativos, mostram que o grafo exibe padrões de mistura

entre ordenamentos perfeitos ou padrões não ordenados, respectivamente [de Oliveira et al., 2021a, Wandelt et al., 2020]. Adicionalmente, as diversas variações de centralidade visam quantificar a importância de cada nó em um grafo. Essa importância assume diferentes sentidos dependendo do tipo de centralidade. Alguns dos principais tipos de métricas clássicas para calcular a centralidade são a proximidade, o grau, a intermediação e o autovetor.

- **Centralidade de Grau** (*Degree*) calcula a importância de um nó a partir do número de arestas conectadas a ele. Assim, a centralidade de grau de um nó é igual à quantidade de arestas que ele possui. Em grafos orientados, o grau do nó é medido em termos de grau de entrada, que considera apenas o número de arestas incidentes no nó, ou grau de saída, que considera apenas o número de arestas partindo do nó. O grau relaciona-se com a popularidade do nó, ou quão bem conectado o nó é. Representa-se a centralidade de grau de um nó v_i por

$$C_{deg}(v_i) = deg(v_i). \tag{4}$$

- **Centralidade de Proximidade** (*Closeness*) relaciona-se à rapidez com que um nó alcança todos os outros nós da rede. O cálculo da centralidade de proximidade ($C_C(v_i)$) de cada nó v_i leva em consideração os caminhos mais curtos entre o v_i e todos os outros nós da rede. Matematicamente, a proximidade é dada por

$$C_C(v_i) = \frac{|\mathscr{V}| - 1}{\sum_{j \neq i} \delta^*(v_i, v_j)}, \tag{5}$$

em que $|\mathscr{V}|$ é o número total de nós e $\delta^*(v_i, v_j)$ é a distância mais curta, em número de saltos, entre o par de nós v_i e v_j. Naturalmente, a inexistência de uma conexão entre v_i e v_j implica uma distância $\delta^*(v_i, v_j) = \infty$. A utilização de caminhos mais curtos permite que a complexidade de tempo para computar a proximidade seja $O(|\mathscr{V}||\mathscr{E}|)$. Na prática, quando usada em um grafo de usuários, a centralidade de proximidade mede efetivamente o quão próximo, em média, cada usuário está de todos os outros usuários em uma comunidade, ou seja, dentro de um grupo de usuários semelhantes que estão densamente conectados.

- **Centralidade de Intermediação** (*Betweenness*) reflete a fração total de caminhos mais curtos que passam por um nó, usando-o como ponte. Dado um grafo $\mathscr{G}$, conexo ou não, a centralidade de intermediação (C_B) do nó v_i é definida como

$$C_B(v_i) = \sum_{v_s \neq v_t \neq v_i \in \mathscr{V}} \frac{\sigma_{st}(v_i)}{\sigma_{st}}, \tag{6}$$

em que $\sigma_{st}(v_i)$ representa o número de caminhos mais curtos do nó v_s para o nó v_t que passam pelo nó v_i e σ_{st} é total de caminhos mais curtos do nó v_s para o nó v_t. Dessa forma, a relação representa a proporção de caminhos mais curtos entre v_s e v_t que passam por v_i. Implicitamente a centralidade de intermediação relaciona-se com o controle de fluxos ou de informação que são propagados pela rede. Quanto mais central é o nó, maior a proporção de caminhos mais curtos que passam por ele

e, consequentemente, maior é o controle exercido sobre os fluxos que trafegam na rede pelos caminhos mais curtos. Realizando buscas em largura, algoritmos podem computar a centralidade de intermediação com uma complexidade de tempo igual a $O(|\mathscr{V}|+|\mathscr{E}|)$.

- **Centralidade de Autovetor** (*Eigenvector*) mede a popularidade de um nó ao considerar a popularidade dos vizinhos desse nó. Dessa maneira, o nó mais central segundo essa centralidade é aquele conectado a mais vizinhos populares. Considerando um grafo $\mathscr{G}$ com um matriz de adjacências $\boldsymbol{A}$, a centralidade de autovetor para o nó v_i é expressa pelo v_i-ésimo elemento do vetor $\boldsymbol{x}$ definido por $\boldsymbol{Ax} = \boldsymbol{\lambda_{max}x}$, em que λ_{max} é o escalar que representa o maior autovalor associado ao autovetor de $\boldsymbol{A}$.

- **Centralidade de Informação** (*Information*) é uma variante da centralidade de proximidade. A centralidade de informação também é chamada de centralidade de proximidade do fluxo de corrente. O valor da centralidade de informação de um nó é determinado com base no fluxo de informação presente em todos os possíveis caminhos entre pares de nós. Essa métrica considera todos os caminhos existentes entre os pares de nós no grafo. Computacionalmente, o cálculo da centralidade de informação é custoso para grandes redes, visto que sua complexidade de tempo possui uma relação cúbica com o numero de nós $O(|\mathscr{V}|^3)$. Matematicamente é definida por

$$C_I(v_i) = \frac{|\mathscr{V}|}{\sum_{v_j \in \mathscr{V}} \frac{1}{I(v_i, v_j)}}, \tag{7}$$

em que $|\mathscr{V}|$ é o número de nós no grafo, $\delta^*(v_i, v_j)$ é o custo do caminho mais curto entre v_i e v_j, e $I(v_i, v_j) = \sum_n I_n(v_i, v_j)$ é o somatório da informação medida em todos os caminhos entre v_i e v_j. A informação em um único caminho n é dada por $I_n(v_i, v_j) = 1/\delta_n(v_i, v_j)$, em que $\delta_n(v_i, v_j)$ é o tamanho do caminho n.

- **Centralidade de Katz** indica a influência relativa de um nó dentro de uma rede medindo o número de vizinhos imediatos e de todos os outros nós da rede que se conectam ao nó em questão por meio desses vizinhos imediatos. Sendo uma generalização da centralidade de autovalor, a centralidade de Katz penaliza as conexões com vizinhos distantes através de um fator de atenuação α, tal que $\alpha < 1/\lambda_{max}$. Matematicamente, a centralidade de Katz de um nó v_i é denotada por

$$C_{Katz}(v_i) = \alpha \sum_{v_j \in \mathscr{V}} a_{j,i} C_{Katz}(v_j) + \beta, \tag{8}$$

em que $a_{j,i}$ é um elemento da matriz de adjacências $\boldsymbol{A}$ de um grafo cujos autovalores são $\boldsymbol{\lambda}$, e o parâmetro β representa um peso relacionado aos vizinhos imediatos. O cálculo dessa centralidade tem a mesma complexidade computacional da centralidade de informação.

- **PageRank** é uma variante da centralidade de autovetor e foi originalmente criada para ranquear a importância de uma página *web*. *PageRank* contabiliza a quantidade e a qualidade de *links* referindo-se à página [Page et al., 1999] para definir a

importância de uma página. A principal medida para estimar a importância do nó é o grau de entrada. Essa métrica é frequentemente incorporada na análise de grafos, uma vez que pode ser utilizada para medir a popularidade de um nó na rede. Diferentemente da centralidade de Katz, nós com elevado *PageRank* não têm influência tão significativa na importância dos seus vizinhos. De maneira simplificada, pode-se calcular o valor de *PageRank* de um nó v_i como

$$C_{PageRank}(v_i) = \alpha \sum_{v_j \in \mathscr{V}} \frac{a_{j,i}}{C_{deg}(v_j)} C_{PageRank}(v_j) + \beta, \tag{9}$$

em que $a_{j,i}$ é um elemento da matriz de adjacências $\boldsymbol{A}$, α é um fator de amortecimento constante, β é um fator de personalização constante e $C_{deg}(v_j)$ é o grau de saída do nó v_j se esse grau for positivo, ou $C_{deg}(v) = 1$ se o grau de saída for nulo. A Equação 9 é definida de forma recursiva, visto que a análise do *PageRank* de determinado nó depende do *PageRank* de todos os nós vizinhos. Destaca-se que a vizinhança entre dois nós exerce maior relevância no cálculo do *PageRank* se o nó de origem possuir maior *PageRank*. Contudo, o *PageRank* de um nó é impactado negativamente proporcionalmente ao número de arestas partindo de si. O *PageRank* mais alto pode ser interpretado como proporcional à probabilidade de o nó espalhar uma informação em sua comunidade.

Visando capturar a qualidade de uma comunidade localmente, a métrica de **condutância** remete à porcentagem de arestas que atravessam os limites de uma comunidade. Comunidades bem definidas tendem a terem uma pequena condutância, significando a densa presença de arestas internas à comunidade, enquanto que arestas externas são esparsas. Assim, dado um grafo $\mathscr{G}$ não direcionado e sua matriz de adjacências $\boldsymbol{A}$, a condutância ϕ de uma comunidade $\mathscr{C} \subset \mathscr{V}$ pode ser calculada pela razão expressa por

$$\phi(\mathscr{C}) = \frac{\sum_{v_i \in \mathscr{C}, v_j \notin \mathscr{C}} a_{i,j}}{\min\left\{a(\mathscr{C}), a(\bar{\mathscr{C}})\right\}}, \tag{10}$$

em que o numerador representa o número de arestas intercomunitárias e o denominador representa o número total de arestas da comunidade. Os valores $a(\mathscr{C})$ são dados por $\sum_{v_i \in \mathscr{C}} \sum_{v_j \in \mathscr{V}} a_{i,j}$ e $a(\bar{\mathscr{C}})$ por $\sum_{v_i \notin \mathscr{C}} \sum_{v_j \in \mathscr{V}} a_{i,j}$. A condutância do grafo é dada pela condutância mínima de todas as possíveis comunidades $\mathscr{C}$, dada por

$$\phi(\mathscr{G}) = \min_{\mathscr{C} \subseteq \mathscr{V}} \phi(\mathscr{C}). \tag{11}$$

A pureza é uma métrica definida pelo produto das frequências dos rótulos mais frequentes associados aos nós de uma comunidade. Cada nó possui um ou mais rótulos que indicam a quais comunidades ele pertence. A pureza de uma comunidade $\mathscr{C}$ é formalizada por

$$P(\mathscr{C}) = \prod_{r \in \mathscr{R}} \frac{\max(\sum_{v_i \in \mathscr{C}} r(v_i))}{|\mathscr{C}|}, \tag{12}$$

em que $\mathscr{R}$ é o conjunto de rótulos, $r \in \mathscr{R}$ é um rótulo e $r(v_i)$ é uma função indicadora que assume valor 1 se $r \in \mathscr{R}(v_i)$.

A partir dos valores de condutância e pureza, Morini *et al.* defendem que o risco de uma comunidade ser efetivamente uma câmara de eco pode ser medido através de uma comparação com dois limiares, p_0 e φ_0, arbitrativamente escolhidos de acordo com o rigor da definição de câmara de eco. Supondo uma escolha adequada dos limiares e o cumprimento das condições $P(\mathscr{C}) > p_0$ e $\phi(\mathscr{C}) < \varphi_0$, pode-se que garantir que: i) a maioria dos usuários da comunidade compartilhe o mesmo rótulo ideológico; ii) a maioria das arestas está dentro dos limites da comunidade. Tais características são frequentemente associadas às câmaras de eco [Morini et al., 2021].

1.6.2. Métricas de Controvérsia em Comunidades

A interpretação de comunidades como câmaras de eco tem como vantagem permitir a aplicação das métricas usadas para análise e detecção de comunidades já existentes. Além disso, essa interpretação possibilita obter indícios da presença das câmaras de eco por meio de duas propriedades: a controvérsia e a homogeneidade. Ambas as propriedades quantificam, de forma complementar, a polarização de um tópico debatido entre usuários de uma rede social. Na prática, a **propriedade de controvérsia** pode ser computada utilizando diversas métricas, tais como *Random Walk Controversy* (RWC), *Authoritative Random Walk Controversy* (ARWC), *Displacement Random Walk Controversy* (DRWC) e *Boundary Connectivity* (BC).

A métrica *Random Walk Controversy* (RWC), originalmente proposta por Garimella *et al.*, considera duas comunidades $\mathscr{X}$ e $\mathscr{Y}$ do grafo $\mathscr{G}$, tal que $\mathscr{X} \cup \mathscr{Y} = \mathscr{V}$, e $\mathscr{X} \cap \mathscr{Y} = \emptyset$. Além disso, são considerados dois passeios aleatórios, um terminando na comunidade $\mathscr{X}$ e o outro terminando na comunidade $\mathscr{Y}$ [Garimella et al., 2018b]. Definida no intervalo $[0,1]$, a RWC expressa a diferença das probabilidades de dois eventos, um considerando que ambos os passeios aleatórios começam e terminam na mesma comunidade, e outro assumindo que ambos os passeios aleatórios começam de uma comunidade e terminam na outra. Matematicamente, essa diferença é computada por

$$RWC = P_{\mathscr{X}\mathscr{X}} P_{\mathscr{Y}\mathscr{Y}} - P_{\mathscr{X}\mathscr{Y}} P_{\mathscr{Y}\mathscr{X}}, \tag{13}$$

em que $P_{\mathscr{A}\mathscr{B}}$ representa a probabilidade condicional de um passeio aleatório começar na comunidade $\mathscr{A}$ e terminar na $\mathscr{B}$, tal que $\mathscr{A}, \mathscr{B} \in \{\mathscr{X}, \mathscr{Y}\}$. Diante disso, espera-se que quanto mais próximo do valor mínimo, maior a probabilidade de migração para a outra comunidade e menor a controvérsia do grafo. Por outro lado, quanto mais próximo do valor máximo, maior a probabilidade de permanência na comunidade original e consequentemente é indicativo da presença de controvérsia.

A métrica ARWC [Villa et al., 2021], proposta por Villa *et al.*, é derivada da RWC, mas distingue-se dela pelo fato de não adotar uma seleção completamente aleatória dos nós iniciais utilizados no passeio aleatório. A ARWC parte apenas dos intitulados nós autoritativos. Tal classificação é atribuída a um conjunto de nós específicos do grafo, com base na centralidade de grau dos mesmos. Essa alteração do ponto de partida permite capturar a probabilidade de um usuário casual, pertencente a uma determinada comunidade, ser exposto ao conteúdo disseminado por um nó autoritativo da comunidade oposta. No

cálculo da ARWC, o passeio aleatório termina quando um nó pertencente ao conjunto de nós autoritativos de uma ou outra partição é alcançado.

Igualmente proposta por Villa *et al.*, a métrica DRWC reflete a razão entre o número de passos, durante um passeio aleatório de comprimento fixo, que resulta em uma mudança de comunidade e o comprimento total do passeio [Villa et al., 2021]. Sendo definida entre $[0,1]$, a métrica é formalmente expressa por

$$DRWC = \frac{\sum_{\forall \upsilon_i \in \mathscr{N}} \left[1 - \left(\frac{n(\upsilon_i)_{cc}}{l_{rw}}\right)\right]}{|\mathscr{N}|}, \tag{14}$$

em que $\mathscr{N}$ representa o conjunto de vértices aleatoriamente selecionados; l_{rw}[10] é o comprimento do passeio aleatório contabilizado em número de arestas; e $n(\upsilon_i)_{cc}$[11] consiste no número de passos no passeio aleatório do nó υ_i, onde o nó migrou de comunidade. Vale ressaltar que, caso nenhuma mudança de comunidade seja identificada durante todo o passeio aleatório de um nó, pode-se inferir que há controvérsia entre as duas comunidades. Não obstante, caso a alternância de comunidade seja frequente, há evidências de que as comunidades apresentam um baixo grau de controvérsia entre si.

Por fim, a métrica BC [Guerra et al., 2013] baseia-se nos conceitos de nós internos e nós limítrofes para mensurar a controvérsia entre duas comunidades. Para pertencer ao conjunto de nós limítrofes $\mathscr{B}_{\mathscr{X}}$ de uma comunidade $\mathscr{X}$, um nó do grafo precisa satisfazer duas condições simultaneamente: i) possuir pelo menos uma aresta conectada a um nó da comunidade oposta $\mathscr{Y}$; ii) possuir pelo menos uma aresta conectada a um membro de $\mathscr{X}$ que não está conectado a $\mathscr{Y}$. Paralelamente, o conjunto de nós da comunidade $\mathscr{X}$ que não pertencem à $\mathscr{B}_{\mathscr{X}}$, são denominados nós internos $\mathscr{I}_{\mathscr{X}}$. Vale ressaltar que, se as duas comunidades constituem câmaras de eco, os nós limítrofes de cada comunidade seriam mais fortemente conectados com os nós internos da mesma comunidade do que com os nós limítrofes da comunidade oposta. A formalização matemática da métrica BC é dada por

$$BC = \frac{1}{|\mathscr{B}|} \sum_{\upsilon_s \in \mathscr{B}} \left[\frac{C^i_{deg}(\upsilon_s)}{C^b_{deg}(\upsilon_s) + C^i_{deg}(\upsilon_s)} - 0{,}5 \right], \tag{15}$$

em que $C^i_{deg}(\upsilon_s)$ representa o número de arestas entre o nó υ_s e os elementos do conjunto $\mathscr{I}$; e $C^b_{deg}(\upsilon_s)$ é o número de arestas entre o nó υ_s e os elementos do conjunto $\mathscr{B}$. Diferentemente das métricas de controvérsia acima descritas, BC é definida entre $[-0.5, 0.5]$, em que valores maiores que zero expressam uma provável presença de controvérsia, enquanto que valores menores que zero indicam falta de polarização. Nesse contexto, quanto maiores os níveis de controvérsia entre nós de diferentes comunidades e de homogeneidade entre nós de uma mesma comunidade, maior a probabilidade da existência de câmaras de eco no grafo analisado [Villa et al., 2021].

1.6.3. Modelagem de Câmaras de Eco e Bolhas de Filtro

Devido à dificuldade de obtenção de bases de dados reais sobre câmaras de eco, alguns trabalhos empregam modelos estatísticos para investigar as câmaras de eco e a

[10] *rw*: *random walk*.

[11] *cc*: *change community*.

propagação de informações e opiniões dentro delas. Embora detenham critérios de propagação particulares, os modelos abaixo podem ser agrupados segundo a forma de representação [Alatawi et al., 2021].

A **Dinâmica Friedkin-Johnson** (FJ) é um modelo de propagação de opinião segundo uma perspectiva de tempo discreto $t \in [0, 1, ..., T]$. O modelo trata a opinião atual de um usuário como uma composição entre uma parcela estática, inerente à própria opinião inata e outra parcela variável no tempo, que é atualizada mediante a influência social. Independentemente do grau de generalização do assunto, específico ou genérico, a dinâmica FJ assume que as opiniões variáveis no tempo podem ser codificadas por um valor real contínuo dentro do espectro polar de $[-1, 1]$. Assim, a completa discordância ou concordância a respeito de um determinado assunto é representada pelos valores extremos do intervalo, respectivamente, enquanto que um valor nulo representa uma opinião nula. Sendo o diferencial do modelo, a intitulada opinião inata é a parcela blindada da influência externa de outros usuários e imutável temporalmente. A construção dessa opinião interna é um resultado histórico, localização geográfica, religião, raça ou outras circunstâncias intrínsecas ao indivíduo. Essa estratégia de modelagem é particularmente útil no estudo de câmaras de eco e bolhas de filtro, quando deseja-se simular a inclinação de diferentes usuários em uma rede social [Chitra e Musco, 2020].

O **Modelo de Bloco Estocástico** (SBM) cria um modelo de grafo de uma comunidade em que os parâmetros do modelo definem o quanto os membros de uma comunidade se misturam dentro e fora de sua comunidade. Assim, o SBM cria um grafo aleatório com base em vários parâmetros. A suposição básica é que existem duas ou mais comunidades criadas a partir de n nós. O SBM é especialmente útil quando combinado com a suposição de homofilia, ou seja, pessoas com inclinações semelhantes tendem a estar conectadas. Deve-se notar, no entanto, que esse modelo não simula homofilia por padrão. O grafo aleatório parametrizado é criado de forma que para cada par de nós, há uma probabilidade p de estarem conectados, dado que estão na mesma comunidade, e há uma probabilidade q de estarem conectados se estiverem em comunidades diferentes. O resultado é um grafo sobre ambas as comunidades. Para simular a homofilia, define-se p maior que q, resultando em nós com as mesmas inclinações, isto é, pertencentes às mesmas comunidades, sendo mais conectados dentro dessas comunidades. O SBM pode ser útil quando combinado com o modelo FJ para estudar os efeitos de bolhas de filtro [Chitra e Musco, 2020]. O modelo FJ é útil para determinar as tendências de equilíbrio dos indivíduos, enquanto o SBM pode fornecer uma maneira de modelar o estado inicial da rede social. Pode-se simular bolhas de filtro adicionando ou removendo arestas [Gausen et al., 2022].

Considerando a hipótese de que no interior das câmaras de eco há prevalência pela divulgação de conteúdos de carácter duvidoso ou falso, pode-se utilizar o modelo de propagação baseado em **Cascata** [Minici et al., 2022]. Nesse modelo, uma estrutura em árvore representa todo o processo de disseminação do conteúdo, podendo ser pautada tanto em uma perspectiva por saltos quanto por tempo. Nessa estrutura, o nó-raiz retrata o primeiro usuário a publicar ou criar um conteúdo e os demais nós representam usuários atuantes no encaminhamento ou compartilhamento do conteúdo falso. A semelhança em cascata entre vários discursos desconexos pode ajudar na detecção da câmara de eco [Tokita et al., 2021].

O modelo de **Confiança Limitada** proposto por Deffuant *et al.* é empregado na predição das opiniões dos usuários com base em suas interações [Deffuant et al., 2000]. Ao representar a opinião de um usuário por um valor dentro do intervalo assimétrico $[0,1]$, o modelo assume que qualquer mudança de opinião está relacionada ao parâmetro de confiança limitado γ. Interpretado como um grau de abertura a novas opiniões, o parâmetro γ define o limite na distância entre a opinião dos dois usuários além do qual a comunicação entre eles não é possível devido a visões conflitantes. Limiares altos geram convergência de opiniões para uma opinião média, ao passo que limiares baixos resultam em vários agrupamentos de opinião, formando, por exemplo, as câmaras de eco. Dessa forma, membros de uma mesma comunidade compartilham a mesma opinião, mas não se ajustam mais com membros de outras comunidades [Sîrbu et al., 2019].

1.7. Representação Vetorial de Textos

Ao optar pela detecção de câmaras de eco segundo a abordagem ideológica, os textos tornam-se a principal matéria-prima de análise. Contudo, para facilitar qualquer análise semântica sobre os conjuntos de dados textuais coletados é essencial que estes sejam primeiramente transformados em estruturas matematicamente operáveis, como matrizes ou vetores numéricos. Essa transformação pode ser desempenhada pelo **Modelo de Espaço Vetorial**, o qual define que textos podem ser interpretados como um espaço vetorial de palavras, em que cada palavra pode ser representada em diferentes padrões. Tais padrões de representação podem embasar-se na: i) contagem binária, em que para cada palavra é atribuído 1 ou 0 de acordo com presença ou ausência da palavra na sentença; ii) contagem cumulativa, usando o Saco-de-Palavras (*Bag-of-Words*), um modelo que computa o número de citações de determinada palavra na sentença; e iii) frequência, usando *tf-idf* (Frequência do Termo–Inverso da Frequência), uma medida estatística que indica a importância de uma palavra de uma sentença em relação a uma coleção de sentenças, ou seja, a base de dados [de Oliveira et al., 2020b]. Apesar da simplicidade e robustez, os modelos clássicos de vetorização tratam as palavras como unidades atômicas, isto é, sem uma conexão semântica entre si. Embora este fator seja menos danoso em uma avaliação da similaridade entre frases ou documentos, esses modelos são incapazes de lidar com a avaliação semântica das palavras. Esse empobrecimento semântico torna palavras com sentidos próximos ou sinônimos totalmente invisíveis à modelagem vetorial. Outra desvantagem consiste na alta dimensionalidade, um reflexo do caráter esparso dos vetores gerados [de Oliveira et al., 2020b, Mikolov et al., 2013].

As dificuldades encontradas em relação às técnicas clássicas de vetorização motivaram o desenvolvimento de técnicas de **incorporações de palavras** (*words embeddings*), uma forma de representação distribuída de palavras. Intimamente ligada a essa forma de representação, a hipótese distribucional manifesta que cada palavra é caracterizada pela sua vizinhança, expressando portanto, uma tendência de palavras com significados semelhantes ocorrerem em contextos similares. Tais representações de palavras podem ser obtidas aplicando modelos preditivos baseados em redes neurais que, quando treinados com grandes volumes de dados textuais, incorporam a semântica das palavras em vetores de baixa dimensão, densos e de tamanho fixo. O principal benefício dessa representação vetorial individualizada para cada palavra encontra-se na preservação das relações semânticas e sintáticas entre palavras. Essa habilidade garante que sinônimos ou

palavras minimamente relacionadas sejam mapeados em vetores similares.A ferramenta *Word2Vec* auxiliou na popularização das técnicas de incorporações de palavras [Mikolov et al., 2013]. Essa ferramenta calcula a representação vetorial de palavras por meio de dois modelos possíveis, o Saco de Palavras Contínuo (*Continuous Bag-of-Words* - CBOW) e o *Skip-gram*, que operam dividindo os textos em dois grupos, palavra-alvo e contexto. O contexto é interpretado como um conjunto limitado das palavras localizado ao redor da palavra-alvo. O tamanho dessa limitação é determinado por uma janela que define o número de palavras a serem consideradas imediatamente à esquerda e à direita da palavra-alvo. O modelo CBOW converge de maneira mais rápida em relação ao *Skip-gram*. Contudo, em relação ao CBOW, o *Skip-gram* apresenta melhores resultados para palavras pouco frequentes. A particularidade do modelo *Skip-gram* está na sua capacidade de usar uma palavra-alvo w_t na predição do contexto de palavras $[w_{t-j}, ..., w_{t+j}]$ em um conjunto de palavras que circunda a palavra-alvo. Já o modelo CBOW inverte a atuação da palavra-alvo e das palavras de contexto, de forma que possibilita a predição de uma palavra-alvo a partir do contexto de palavras próximas.

A Figura 1.4 mostra a arquitetura do modelo *Skip-gram*. A arquitetura é composta pelas camadas de entrada e de saída, intercaladas por uma camada oculta. O número de palavras V existentes no vocabulário usado no treinamento determina o tamanho das camadas de entrada e de saída. Já o tamanho da camada oculta é determinado com base em um parâmetro N arbitrário, que expressa a dimensão do futuro vetor de palavras gerado H (*word embeddings*). Essa dimensão indica a quantidade de características usadas na representação numérica de cada palavra, sendo, portanto, inferior à dimensão do vetor original de cada palavra inserido na camada de entrada. A conexão da camada de entrada para a camada oculta é feita através de uma matriz de pesos W_I de tamanho $V \times N$. Analogamente, a conexão da camada oculta para a camada de saída é desempenhada pela matriz W_O de tamanho $N \times V$. Ambas as matrizes de peso W_I e W_O são inicializadas com valores aleatórios pequenos. A inserção de uma palavra-alvo na camada de entrada da rede neural inicia com a codificação dessa palavra em seu vetor *one-hot*, uma matriz coluna $V \times 1$ usada para distinguir cada palavra em um vocabulário. Esse vetor consiste em 0s em todas as posições, com exceção de um único 1 na posição x_i usada exclusivamente para identificar a palavra [de Oliveira et al., 2021b].

No processo de treinamento, a cada iteração são empregados dois algoritmos de aprendizado: de propagação direta (*forward propagation*) e de retropropagação (*back-propagation*). Aplicando primeiramente o algoritmo de propagação direta, o vetor *one-hot* da palavra-alvo de entrada é multiplicado pela matriz de pesos W_I para formar o vetor H da camada oculta. Em seguida, o vetor H é então multiplicado por W_O gerando assim C vetores intermediários idênticos, cada um representando uma palavra de contexto. As saídas do modelo são adquiridas aplicando a cada vetor intermediário a função *softmax*. Esta função tem o objetivo de normalizar o vetor intermediário U composto por V números flutuantes, transformando-o no vetor de distribuição de probabilidade Y. Uma vez descoberto o vetor normalizado de probabilidades de cada palavra de contexto, o algoritmo de retropropagação os compara com o vetor *one-hot* da palavra correspondente para assim atualizar as matrizes de peso W_I e W_O. Essa atualização ocorre especificamente nos valores da coluna correspondente de W_O e da linha correspondente de W_I. No modelo CBOW, devido à inversão da atuação da palavra-alvo e das palavras de contexto, admitem-se múl-

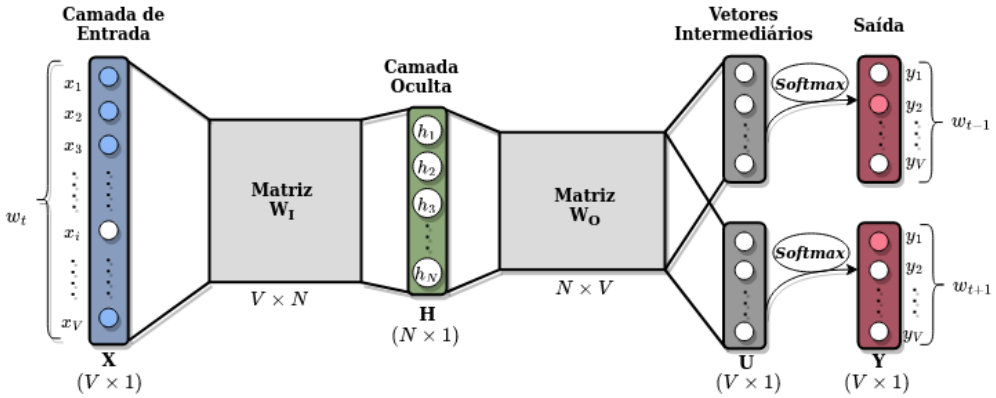


Figura 1.4. Arquitetura do modelo *Skip-gram* considerando como entrada a palavra-alvo w_t codificada no seu vetor *one-hot* X. Na saída do modelo são obtidos C vetores de distribuição de probabilidade, um para cada palavra do contexto. Com o modelo devidamente treinado, espera-se que as maiores probabilidades de cada vetor Y, encontradas nas posições y_2 e y_1, expressem as palavras de contexto w_{t-1} e w_{t+1} relacionadas a palavra alvo. Adaptado de [de Oliveira et al., 2021b].

tiplas entradas, uma para cada palavra de contexto. Devido à multiplicidade de vetores de entrada é necessário calcular a média dos vetores de palavras correspondentes, construídos pela multiplicação dos múltiplos vetores *one-hot* de entrada e pela matriz W. Uma segunda consequência da inversão é a presença de uma única função *softmax*, ao contrário das C existentes na arquitetura do modelo *Skip-gram* [de Oliveira et al., 2021b].

1.8. Representação Vetorial de Grafos

A adoção da abordagem topológica para a identificação de câmaras de eco implica a necessidade de modelar a rede como um grafo. A transformação de um grafo em uma representação vetorial auxilia na avaliação quantitativa da qualidade da estrutura da comunidade gerada. Na representação vetorial de grafos, os nós do grafo são mapeados em um espaço vetorial derivado. Goyal e Ferrara argumentam que a obtenção de representação vetorial de grafos é uma tarefa inerentemente desafiadora e precisa atentar-se a três aspectos [Goyal e Ferrara, 2018]:

- **Preservação de Propriedades**. A qualidade de uma representação vetorial de grafos está atrelada à sua capacidade de preservar as propriedades estruturais da conexão entre os nós individuais. Esse grau de preservação pode ser delimitado segundo três tipos de medidas de proximidade. A *proximidade de primeira ordem* visa quantificar a similaridade local entre os nós a partir do peso da aresta que os interliga. A *proximidade de segunda ordem* captura a proximidade entre as estruturas de vizinhança dos nós. Tal proximidade pode ser facilmente estimada usando a métrica de probabilidade de transição entre dois nós. Embora exista a *proximidade de alta ordem*, essa tem pouca ocorrência na literatura uma vez que a proximidade de segunda ordem supre satisfatoriamente a necessidade da maioria dos métodos de incorporação de grafos.
- **Escalabilidade**. Caracterizadas pela grande quantidade de nós e arestas, as redes

sociais têm seu processamento e análise diretamente impactados pelo modelo de representação vetorial adotado. A aplicação de métodos tradicionais nessas redes gera representações vetoriais demasiadamente extensas, com dimensão proporcional ao número de nós. Esse aspecto compromete a aplicabilidade de métodos de representações vetoriais às redes reais de grande escala. Na prática, a definição de métodos de vetorização escaláveis é especialmente difícil quando se deseja preservar as propriedades globais da rede.

- **Dimensionalidade**. O dimensionamento ótimo de representações vetoriais de grafos é uma tarefa subjetiva. Embora a escolha de um número maior de dimensões possa favorecer precisão em tarefas de reconstrução do grafos, isso também pode aumentar a complexidade computacional de procedimentos ou algoritmos subsequentes.

1.8.1. Matriz de Adjacências

A matriz de adjacências apresenta-se como uma das estratégias mais simples e tradicionais de vetorização de grafos. A matriz de adjacência permite representar um grafo $\mathscr{G}$ composto por $|\mathscr{V}|$ vértices por meio de uma matriz $\boldsymbol{A}(\mathscr{G}) = [a_{ij}]$ de dimensão $n \times n$, em que os elementos a_{ij} da matriz dependem das propriedades intrínsecas ao grafo. Supondo um grafo não ponderado, seja direcionado ou não direcionado, cada elemento a_{ij} da matriz $\boldsymbol{A}$ contém 1, caso v_i e v_j sejam adjacentes e 0 caso contrário. Em grafos ponderados, os elementos a_{ij} retratam o peso da aresta entre v_i e v_j. Essa lógica de preenchimento garante que grafos não direcionados sejam mapeados em matrizes de adjacência simétricas ao longo da diagonal principal. Todavia, tal simetria não é garantida em grafos direcionados. Embora o vetor de adjacências de um nó codifique a estrutura de vizinhança de primeira ordem de um nó, o vetor resultante é esparso, discreto e de alta dimensionalidade devido à natureza esparsa de redes de grande escala [Cui et al., 2019, Xu, 2021].

1.8.2. Incorporação de Grafos

Na literatura, o termo *incorporação de grafos* é descrito como uma forma de representar um grafo inteiro, ou cada nó individualmente, em um espaço vetorial de menor dimensão que o original. Assim, o principal objetivo da incorporação de grafos baseados em pontos vetoriais consiste em projetar nós de grafos de alta dimensão em vetores de baixa dimensão em um espaço latente. Nessa transformação deve-se garantir a preservação das propriedades originais da estrutura do grafo, de forma que nós próximos entre si no grafo original sejam posteriormente incorporados a um espaço latente com representações vetoriais semelhantes [Xu, 2021]. Os métodos de incorporação de grafos podem ser categorizados em três tipos principais baseados em i) fatoração de matrizes; ii) passeio aleatório; e iii) aprendizado profundo. A Tabela 1.4 sumariza as características de alguns métodos pertencentes a cada tipo. A complexidade de diversos algoritmos mostrados na tabela é dependente da dimensão d escolhida para a incorporação. Esta seção discute os métodos e algoritmos enumerados na tabela.

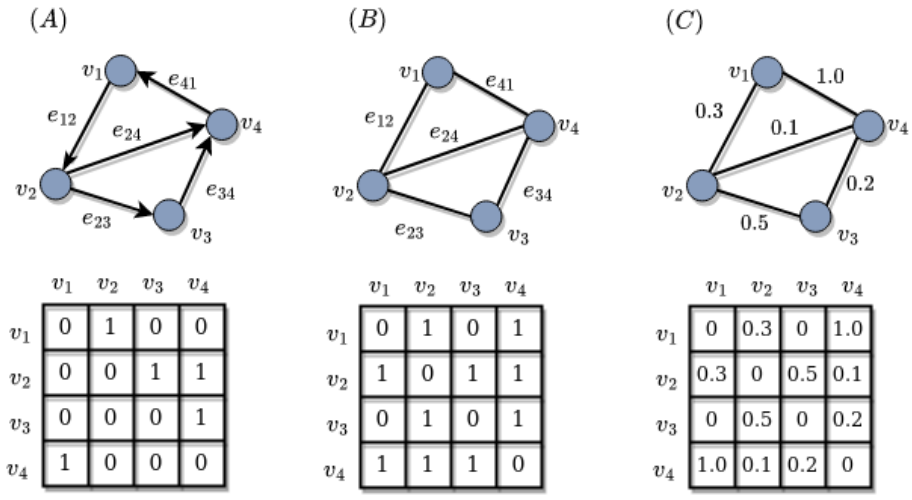


Figura 1.5. Diferentes tipos de grafos e suas respectivas representações matriciais de adjacência. A primeira linha de (*A*) a (*C*) são, respectivamente, exemplos de grafos direcionados, não direcionados e ponderados. A principal diferença entre (*A*) e (*B*) é que as arestas são direcionadas em (*A*), mas não direcionadas em (*B*). (*C*) mostra um grafo ponderado onde cada aresta é ponderada com um valor específico. As matrizes de adjacência 4×4 correspondentes a cada grafo encontram-se imediatamente abaixo dos mesmos. Convém ressaltar que como nenhum grafo possui auto-laços (*self-loops*), *i.e.*, arestas que se originam e terminam no mesmo nó, as matrizes de adjacência de cada grafo têm todos os elementos da diagonal principal nulos.

Incorporação Baseada na Fatoração de Matrizes

Os métodos de incorporação de grafos baseados em fatoração representam as conexões entre nós na forma de uma matriz e fatorizam essa matriz para obter a incorporação. Dentre os tipos de matrizes utilizáveis para representar as conexões estão a matriz de adjacências de nós, matriz Laplaciana, matriz de probabilidade de transição de nós e matriz de similaridade de Katz. Dependendo das propriedades originais da matriz, adota-se uma abordagem de fatoração específica. Nesse contexto, matrizes positivas semidefinidas, e.g., a matriz Laplaciana, podem ser submetidas à decomposição de autovalores, ao passo que matrizes não estruturadas podem ser incorporadas segundo métodos de gradiente descendente [Goyal e Ferrara, 2018]. Outra técnica comumente utilizada na fatoração de matrizes é a Decomposição de Valor Singular (SVD). A partir da seleção de um nível de aproximação k, a SVD encontra uma versão aproximada da estrutura matricial do grafo original, omitindo todos, exceto os k maiores valores singulares na decomposição.

O algoritmo **Automapas Laplacianos** [Belkin e Niyogi, 2001] parte da premissa que nós interligados por arestas fortemente ponderadas devem possuir incorporações semelhantes. Para atingir esse objetivo, o algoritmo inicia construindo um grafo seguindo uma abordagem baseada na ε-vizinhança ou uma abordagem baseada em K-vizinhos mais próximos. Embora seja geometricamente motivada, a primeira abordagem pode frequentemente gerar muitas componentes conectadas dependendo do valor do parâmetro ε. Tal parâmetro reflete o limiar máximo da distância entre dois nós para ambos que sejam interligados por uma aresta. Independentemente da escolha da abordagem de constru-

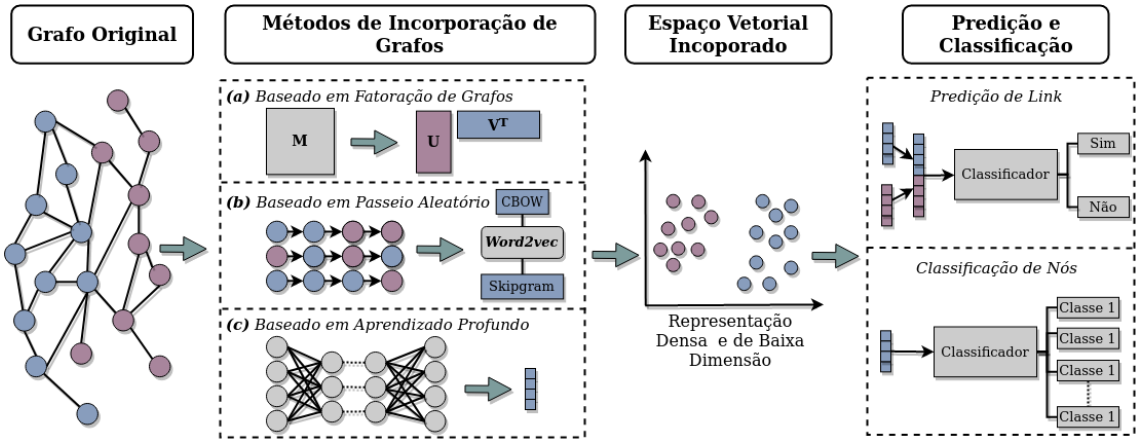


Figura 1.6. Visão genérica do processo de incorporação de grafos. Partindo de um grafo previamente estruturado, este pode ser submetido a três tipos de métodos de incorporação, sendo eles: (A) baseados em fatoração de matrizes, que aplicam técnicas de redução dimensional em representações matriciais do grafo original; (B) baseados em passeio aleatório, que geram sequências aleatórias de nós para nutrir modelos de incorporação *e.g.*, *Word2vec*; e (C) baseados em aprendizado profundo, que utilizam rede neurais com arquiteturas e entradas específicas. Após obter uma representação vetorial densa e de baixa dimensão sobre nó do grafo, pode-se construir classificadores específicos para diferentes aplicações. Adaptado de [Yue et al., 2020].

ção, o algoritmo posteriormente atribui pesos às arestas utilizando ponderação unitária ou calculando-os através da função *heat kernel*[12]. Por fim, o algoritmo computa os automapas gerando as representações em baixa dimensão de cada nó. O caráter de preservação de localidade do algoritmo, torna-o relativamente insensível a pontos discrepantes (*outliers*) e ruídos. Um subproduto dessa preservação é que o algoritmo enfatiza implicitamente as comunidades naturais nos dados.

A **Fatoração de Grafos** (GF) [Ahmed et al., 2013] é um método que propõe uma incorporação de grafos por meio da fatoração da matriz de adjacências do grafo de entrada. Esse método aplica uma técnica de fatoração baseada em particionamento do grafo, visando minimizar o número de nós vizinhos em vez de arestas entre as partições. Internamente, a função objetivo do método contém um coeficiente de regulação intuitivamente ajustável e capaz de controlar a generalização da incorporação gerada. Um coeficiente de regularização baixo propicia uma melhor reconstrução, porém pode eventualmente particularizar, levando a um desempenho de previsão ruim. Por outro lado, a escolha de um coeficiente demasiadamente alto põe em uma vez que resulta numa sub-representação os dados [Goyal e Ferrara, 2018].

Proposto por Ou *et al.*, o método **HOPE** (*High-Order Proximity preserved Embedding*) visa preservar a propriedade de transitividade assimétrica, uma propriedade crítica em redes direcionadas [Ou et al., 2016]. A transitividade assimétrica retrata a correlação entre arestas, indicando que se houver um caminho direcionado de v_i para v_j, então provavelmente há uma aresta direcionada de v_i para v_j. Para preservar esta propriedade, o HOPE constrói uma formulação geral de quatro medidas de proximidade de alta or-

[12]Função dada por $h_t = e^{-\frac{||v_i - v_j||^2}{t}}$, em que t é o tempo e v_i e v_j são os nós conectados por uma aresta.

Tabela 1.4. Principais Métodos e Algoritmos de Incorporação de Grafos.

Categoria	Referência	Método ou Algoritmo	Complexidade de Tempo	Propriedades Preservadas
Fatoração	[Roweis e Saul, 2000]	LLE	$O(\|\mathscr{E}\|d^2)$	Proximidade de 1ª ordem
	[Belkin e Niyogi, 2001]	Automapas Laplacianos	$O(\|\mathscr{E}\|d^2)$	
	[Ahmed et al., 2013]	Fatoração de Grafo	$O(\|\mathscr{E}\|d)$	
	[Cao et al., 2015]	GraRep	$O(\|\mathscr{V}\|^3)$	Proximidades de 1-kª ordem
	[Ou et al., 2016]	HOPE	$O(\|\mathscr{E}\|d^2)$	
Passeio Aleatório	[Perozzi et al., 2014]	DeepWalk	$O(\|\mathscr{V}\|d)$	
	[Grover e Leskovec, 2016]	Node2vec	$O(\|\mathscr{V}\|d)$	Proximidade de 1-kª , equivalência estrutural
Aprendizado Profundo	[Cao et al., 2016]	DNGR	$O(\|\mathscr{V}\|^2)$	Proximidades de 1-kª ordem
	[Welling e Kipf, 2016]	GCN	$O(\|\mathscr{E}\|d^2)$	
	[Wang et al., 2016]	SDNE	$O(\|\mathscr{V}\|\|\mathscr{E}\|)$	Proximidades de 1ª e 2ª ordem
Miscelânea	[Tang et al., 2015]	LINE	$O(\|\mathscr{E}\|d)$	

dem. Posteriormente, aplica-se o SVD generalizado à formulação a fim de encontrar uma versão aproximada e compacta. Esse processo de redução dimensional diminui significativamente a complexidade de tempo do método [Cui et al., 2019]. O HOPE fornece um limite superior teórico para o erro de aproximação e, assim, consegue estimar a qualidade da incorporação e determinar as dimensões de incorporação automaticamente.

Incorporação Baseada em Passeio Aleatório

A tentativa de transpor a tarefa de incorporação de um cenário linguístico para um cenário em grafos, depara-se com a indeterminação de como delimitar adequadamente a vizinhança em grafos. Para contornar essa indeterminação, alguns métodos de incorporação de grafos baseiam-se no conceito de passeio aleatório para amostrar aleatoriamente os vizinhos do nó e assim extrair o contexto estrutural do nó. A aplicabilidade do passeio aleatório na incorporação de grafos torna-se mais evidente quando o grafo está parcialmente disponível ou é demasiadamente grande para ser analisado em sua totalidade [Goyal e Ferrara, 2018]. Estabelecendo um paralelo com o cenário textual, cada nó integrante de um grafo é tratado como uma palavra individual, enquanto que um passeio aleatório é interpretado como uma sentença [Cui et al., 2019]. Além do processo de amostragem com passeio aleatório, os métodos de incorporação baseados em passeio aleatório costumam incluir um modelo de incorporação tipicamente usados para linguagem, como o *SkipGram*. Tais modelos atuam como codificadores eficazes para problemas prevalentes

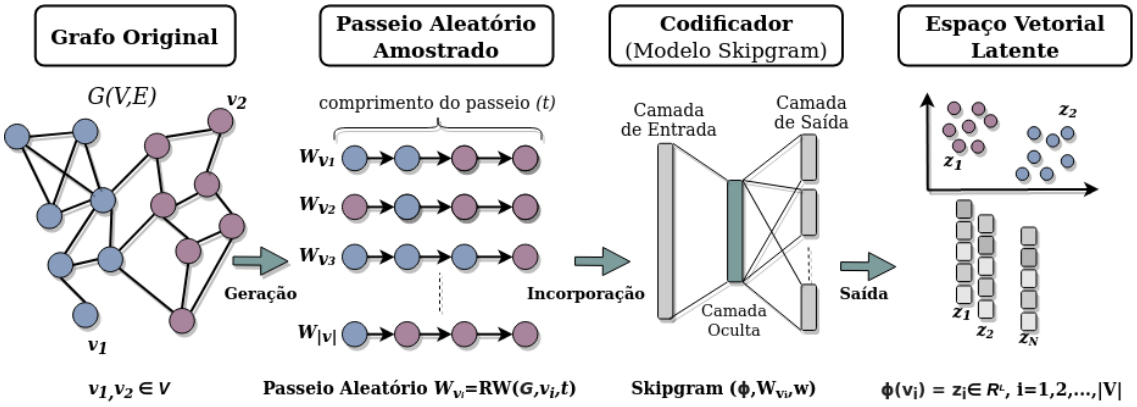


Figura 1.7. Processo de incorporação de grafos implementado por métodos baseados em passeio aleatório. Em posse do grafo de entrada, os métodos inicialmente aplicam um procedimento de amostragem de nós baseada no passeio aleatório. Cada passeio aleatório parte de um nó do grafo e tem um comprimento em saltos fixo igual a t. Assim, cada passeio aleatório pode ser representado pelo contexto do nó correspondente. Em seguida , um modelo de incorporação de linguagem desempenha o papel de um codificador de modo que cada nó seja representado como um vetor contínuo de baixa dimensão no espaço latente. Nesses vetores estão preservadas as propriedades estruturais do grafo original. Adaptado de [Xu, 2021].

de incorporação de nós de grafos. Dessa forma, os métodos baseados em passeio aleatório conseguem codificar efetivamente a estrutura e as informações topológicas do grafo original no espaço latente.

O algoritmo **Node2Vec**[13] integra o modelo *Skip-gram* em seu funcionamento. O *Node2Vec* permite mapear nós em representações vetoriais densas e de baixa dimensão [Grover e Leskovec, 2016]. Nesse espaço de incorporação, garante-se a preservação da estrutura de comunidade, bem como a equivalência estrutural entre nós do grafo original. Para preservar a proximidade de ordem superior entre nós, o *Node2vec* maximiza a probabilidade de ocorrência de nós subsequentes em passeios aleatórios de comprimento fixo [Cui et al., 2019, Goyal e Ferrara, 2018]. Diferentemente do *DeepWalk*, o *Node2vec* emprega passeios aleatórios tendenciosos que fornecem uma troca entre buscas em largura (*Breadth-First Search* - BFS) e em profundidade (*Depth-First Search* - DFS). A consequência disso é a produção de incorporações de maior qualidade e mais informativas comparadas às do *DeepWalk*.

Incorporação Baseada em Aprendizado Profundo

A incorporação de grafos é inerentemente uma tarefa que transforma o espaço original em um espaço vetorial de baixa dimensão. O desafio intrínseco a essa tarefa está justamente no aprendizado de uma função de mapeamento entre esses dois espaços. Na tentativa de resolver esse desafio, os métodos de incorporação baseados em fatoração de matrizes assumem que a função de mapeamento é linear. No entanto, diante da complexidade do processo de formação de um grafo, não há garantias que esse processo seja linearmente modelável. Sendo assim, uma função linear pode não ser suficientemente

[13]Disponível em https://github.com/aditya-grover/node2vec.

capaz de mapear o grafo original para um espaço de incorporação. Nesse contexto, as redes neurais profundas apresentam-se como uma solução eficaz para aprender funções não lineares de incorporação de grafos. Os principais desafios são o ajuste de modelos profundos aos dados da rede e a imposição da estrutura da rede e as restrições de nível de propriedade aos modelos profundos [Cui et al., 2019].

Idealizado por Wang *et al.*, o **SDNE** (*Structural Deep Network Embedding*) emprega um modelo profundo semi-supervisionado para preservar as proximidades de primeira e segunda ordem de um grafo [Wang et al., 2016]. O modelo é composto por várias camadas de funções não lineares e pode ser dividido em uma componente não supervisionada e outra supervisionada. Internamente, a componente supervisionada explora a proximidade de segunda ordem para preservar a estrutura global do grafo, reconstruindo a estrutura de vizinhança de cada nó. Enquanto isso, a componente supervisionada explora a proximidade de primeira ordem para preservar a estrutura local do grafo. Essa componente supervisionada é baseada no algoritmo de automapas laplacianos que aplica uma penalidade quando vértices semelhantes são mapeados longes um do outro no espaço de incorporação. Tais características garantem a robustez necessária para que o SNDE lide com redes esparsas.

Indicado para incorporação em redes de larga escala, o **LINE** define explicitamente duas funções [Tang et al., 2015]. A primeira função, destinada à preservação da proximidade de primeira ordem, mede a similaridade entre pares de nós através da distribuição de probabilidade conjunta entre eles. Essa primeira função assemelha-se com aquela adotada no método de Fatoração de Grafos, uma vez que ambas visam manter próximos a matriz de adjacências e o produto escalar das incorporações. Paralelamente, a segunda função visa preservar a proximidade de primeira ordem medindo a semelhança dos vizinhos, *i.e.*, o contexto de dois nós. A distribuição condicional implica que nós com distribuições semelhantes nos contextos sejam semelhantes entre si. Ao minimizar a divergência de Kullback-Leibler das duas distribuições e das distribuições empíricas, pode-se obter as representações de nós que são capazes de preservar as proximidades de primeira e segunda ordem [Cui et al., 2019, Goyal e Ferrara, 2018].

1.9. Delimitação e Análise de Câmaras de Eco em Redes Sociais

As câmaras de eco são estruturas que têm o potencial de ampliar as ideias disseminadas dentro da própria estrutura ao mesmo tempo em que isola essas ideias de refutação. Essas estruturas são formadas por usuários altamente conectados. A comunidade em uma rede social também é uma estrutura altamente conectada, formada por um subconjunto de nós da rede que possuem maior densidade de conexões entre eles do que com o restante dos nós da rede. Assim, estruturalmente, as câmaras de eco podem ser interpretadas como comunidades em um grafo, de forma que algoritmos de detecção de comunidade amplamente difundidos na literatura podem ser empregados para identificar e delimitar as câmaras de eco. A qualidade das comunidades encontradas é avaliada por meio de métricas de avaliação extrínsecas e intrínsecas. Esta seção apresenta diversos algoritmos utilizados para detecção de comunidade e métricas de avaliação que podem ser aplicados para delimitação e análise de câmaras de eco em redes sociais.

1.9.1. Algoritmos de Detecção de Comunidade

Os algoritmos de detecção de comunidade são capazes de desvendar a existência de uma organização de rede interna não trivial de modo geral. Isso permite inferir relações especiais entre os nós que podem não ser facilmente acessíveis a partir de testes empíricos diretos e ajuda a entender melhor as propriedades dos processos dinâmicos que ocorrem em uma rede [Yang et al., 2016]. Embora cada algoritmo detenha uma lógica própria, todos convergem para um objetivo comum de encontrar subgrafos com características estruturais homogêneas. Sendo a métrica base de diversos algoritmos, a modularidade reflete a tendência de formar agrupamentos que podem ser interpretados como comunidades, chamados de módulos da rede. Pode-se definir a modularidade (Q) de um grafo conexo por

$$Q = \frac{1}{2|\mathscr{E}|} \sum_{i,j} \left[a_{i,j} - \frac{C_{deg}(\upsilon_i) C_{deg}(\upsilon_j)}{2|\mathscr{E}|} \right] \gamma(c_i, c_j) \tag{16}$$

em que $|\mathscr{E}|$ é número total de arestas do grafo, $C_{deg}(\upsilon_x)$ é o grau do nó υ_x, que em um grafo ponderado é o peso total das arestas conectadas ao nó υ_x, $a_{i,j}$ é o peso da aresta entre υ_i e υ_j. O nó υ_x faz parte da comunidade c_x e há uma função indicadora $\gamma(c_i, c_j)$ que assume valor unitário, se $c_i = c_j$, e é nula, se $c_i \neq c_j$, ou seja, a função indicadora é igual a 1 se os nós υ_i e υ_j estão na mesma comunidade. A Figura 1.8 mostra a variação da modularidade de um grafo conexo, considerando diferentes cenários de segregação em comunidades. Em (A), o surgimento de duas comunidades bem definidas e fracamente conectadas entre si, resulta em um valor ótimo de modularidade. Em compensação, em (B) observa-se um cenário subótimo de divisão em comunidades. Em (C), constata-se a existência de uma única comunidade contendo todos os nós do grafo, implicando uma modularidade nula. Por fim, a modularidade negativa vista em (D) indica que cada nó do grafo é uma comunidade em si. Observa-se que quanto maior a modularidade da rede, maior a tendência de formar comunidades com forte conectividade entre os membros. A modularidade é capaz de medir o quão forte é a conexão de um nó adicionado a uma comunidade em contraste à sua adição a uma comunidade aleatória. Outros algoritmos empregam o passeio aleatório (*random walk*). No passeio aleatório, a sequência de transições entre nós de um grafo é modelada por uma Cadeia de Markov finita e temporalmente reversível.

Algoritmo *Louvain*

O algoritmo de *Louvain* é uma abordagem heurística não supervisionada que visa maximizar a modularidade a partir de sucessivas redistribuições de nós entre as várias comunidades de um grafo. Assim, o algoritmo Louvain divide-se em 2 fases, otimização da modularidade e agregação de comunidade. Após a otimização da modularidade executa-se a agregação de modularidade. Na fase de otimização da modularidade, o algoritmo ordena os nós aleatoriamente e verifica a modularidade ao remover cada nó de uma comunidade e adicioná-lo em outra comunidade, até que não haja mais um aumento significativo na modularidade. Na fase da agregação de comunidade, todos os nós que pertencem a uma mesma comunidade são fundidos em um único nó representativo dessa comunidade, o nó gigante. Os enlaces que conectam nós gigantes são o conjunto dos enlaces que conectavam os nós que fazem parte do nó gigante. Isso pode gerar auto-laços

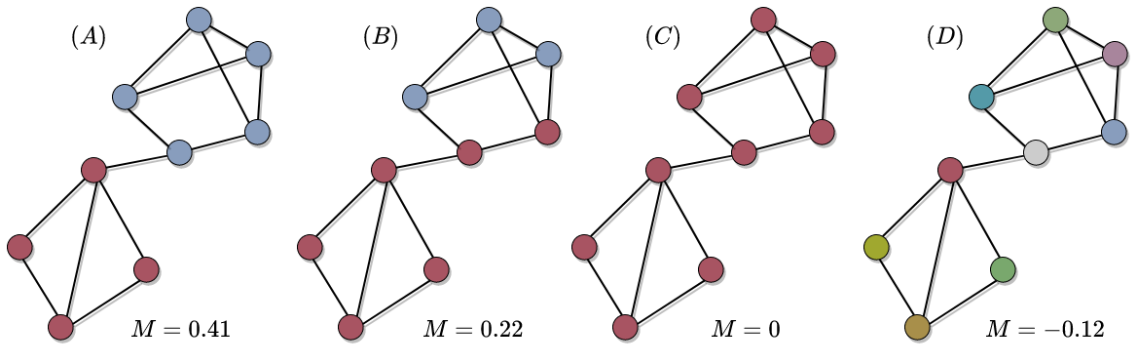


Figura 1.8. Variação da modularidade de um grafo conexo, considerando diferentes cenários de segregação em comunidades. As cores dos nós representam a comunidade à qual estão associados. A modularidade diminui do cenário (*A*) em direção ao cenário (*D*). Quanto maior a modularidade da rede, maior a tendência de formar comunidades com forte conectividade entre os membros.

(*self-loops*) conectando o nó gigante a ele mesmo. Essas duas fases são repetidas até alcançar a convergência, quando nenhum outro remanejamento de nós proporciona um aumento na modularidade total do grafo. A popularidade do algoritmo é reflexo da sua eficiência, sendo capaz de garantir a rápida construção de comunidades mesmo em redes extremamente populosas [Alatawi et al., 2021].

Fast Greedy

Implementando uma lógica baseada no agrupamento hierárquico aglomerativo, o *Fast Greedy* [Clauset et al., 2004] busca segregar o grafo original em comunidades a partir da otimização da métrica de modularidade. Ao optar pelo método aglomerativo, *i.e. bottom-up*, o algoritmo inicia atribuindo uma comunidade única a cada nó do grafo. Guiando-se pelas potenciais alterações na modularidade do grafo, o *Fast Greedy* elege as duas comunidades a serem mescladas de cada iteração. Na prática, o par que fornece o máximo de melhoria de modularidade é selecionado para compor uma nova comunidade. Tal procedimento é repetido até que nenhuma fusão de pares de comunidade culmine em um aumento na modularidade. A eficiência do algoritmo reflete em um tempo de execução linear mesmo em redes extremamente grandes. Caso a rede analisada apresente uma estrutura esparsa e hierárquica, o algoritmo pode atingir uma complexidade de tempo de $O(|\mathcal{V}|log^2|\mathcal{V}|)$, em que $\mathcal{V}$ é o número de nós.

Algoritmo de Propagação de Rótulos

Igualmente rápido na geração de comunidades, o Algoritmo de Propagação de Rótulos (*Label Propagation Algorithm* - LPA) prevê que cada nó deve ser atribuído à mesma comunidade que a maioria de seus vizinhos diretos. Para implementar tal estratégia, o algoritmo normalmente inicializa alocando um rótulo distinto a cada nó do grafo. Em seguida, segundo uma ordem aleatoriamente gerada de nós, o algoritmo executa um processo iterativo em que cada nó recebe o rótulo predominante na sua vizinhança. Essa

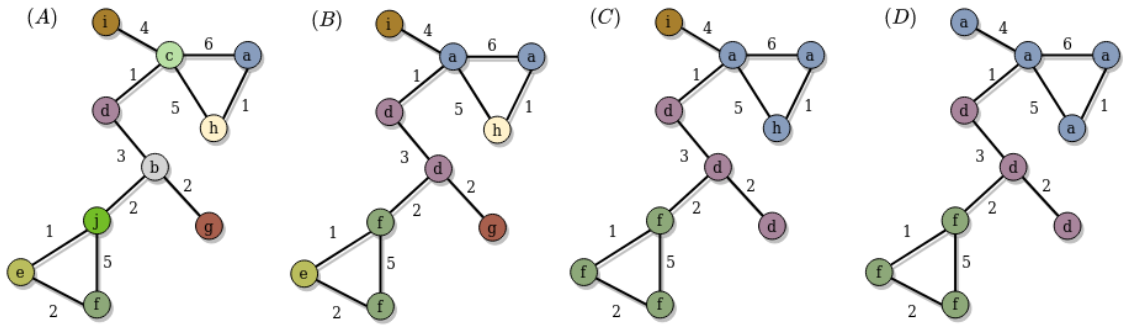


Figura 1.9. Princípio de funcionamento do LPA sobre um grafo ponderado. No cenário (*A*), o algoritmo inicia distribuindo rótulos únicos a cada nó pertencente ao grafo. A cada iteração, o LPA escolhe e processa de forma aleatória cada nó do grafo, atribuindo a nós o rótulo de seu vizinho com o peso máximo. Esse processo de atualização dos rótulos ocorre entre as várias iterações entre os cenários (*B*) e (*C*). A convergência observada no cenário (*D*) ocorre quando todos os nós obtêm o rótulo majoritário de seus vizinhos ou quando alcança-se o limite máximo de iterações pré-estabelecido.

atualização do rótulo do nó depende do peso máximo, calculado com base nos pesos dos nós vizinhos e seus relacionamentos. Após a convergência do processo, grupos de nós com o mesmo rótulo são interpretados como comunidades [Yang et al., 2016]. Dessa forma, dependendo da ponderação das arestas e do grau de conexão de um nó, seu rótulo pode rapidamente tornar-se dominante em um grupo de nós densamente conectado. Do mesmo modo, este rótulo terá dificuldades para cruzar uma região escassamente conectada.

É possível haver iterações durante a execução do algoritmo em que nós selecionados não tenham seus rótulos atualizados. Isso ocorre devido aos vizinhos com o peso máximo já possuírem o mesmo rótulo que o nó em questão. Além disso, ocorrem empates, ou seja, vizinhos com peso máximo iguais e rótulos diferentes. O empate é resolvido de maneira uniforme e aleatória. Devido a eventuais problemas de convergência, é aconselhável que o LPA seja implementado especificando um limite máximo de iterações. A configuração deste parâmetro evita a ocorrência de ciclos infinitos de trocas de rótulos que impactariam na eficiência do algoritmo. Diferentemente de outros algoritmos, o LPA pode retornar diferentes estruturas de comunidade quando executado várias vezes sob o mesmo grafo. Essa variabilidade de resultados possíveis é influenciada pela ordem com que o LPA avalia os nós, bem como pelo processo de desempate aleatório.

Método de Percolação de Cliques

O Método de Percolação de Cliques (*Clique Percolation Method* - CPM) concentra-se na detecção de comunidades interpretando-as como subgrafos totalmente conectados, os k-cliques. A partir de um valor inteiro pré-estabelecido (k), o método identifica cliques de k nós e os aglutina caso dois cliques compartilhem $k-1$ nós. Esse processo de identificação e aglutinação de cliques é repetido até que não haja mais junções possíveis [Alsini et al., 2020]. A Figura 1.9.1 mostra exemplos de k-cliques. É

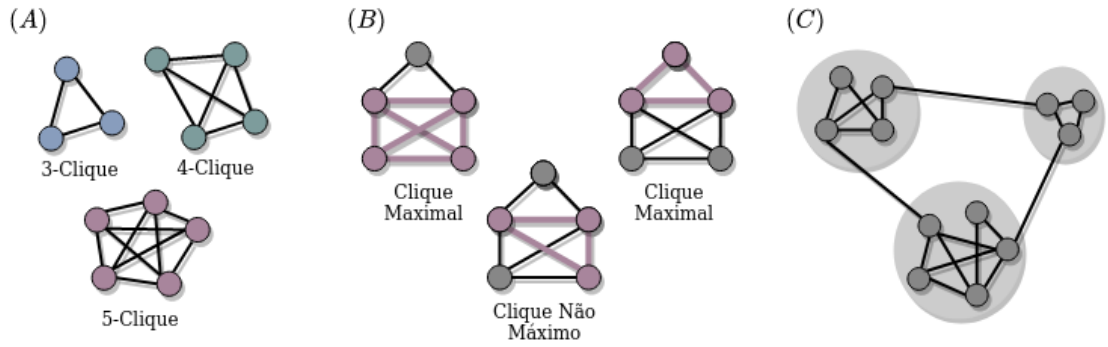


Figura 1.10. Em (*A*) ilustra-se vários tipos de k-clique. Em (*B*) mostra-se as diferenças entre clique máximo e maximal em um mesmo grafo. Na prática, um clique máximo é o clique que inclui o maior número possível de nós do grafo. Já um clique maximal é clique que não pode ser aumentado pela inclusão de mais um nó, alcançando assim ápice do seu tamanho. Assim todo clique máximo é maximal, mas o contrário não. Em (*C*) exemplifica-se as comunidades geradas pelo CPM utilizando $k = 3$. Neste exemplo, o CPM agrupa triângulos (3-clique) mesma comunidade quando estes têm $k-1$ nós em comum.

importante destacar e diferenciar os conceitos de clique maximal e clique máximo. No processo de aglutinação de cliques, eventualmente não é mais possível aumentar o tamanho de um determinado clique porque ele não é um subconjunto de um clique maior. Um clique máximo é o maior clique da rede, isto é, aquele que possui a maior quantidade de nós. Em decorrência da capacidade de gerar comunidades sobrepostas, o CPM é útil na representação realista de redes sociais, visto que usuários reais podem pertencer a várias comunidades simultaneamente. Em redes sociais reais, a não exclusividade de pertencimento vale tanto para núcleos familiares ou de amizade, quanto para câmaras de eco relacionadas a temas distintos [Alduaiji et al., 2018]. A lógica simples do CPM garante também uma rapidez no funcionamento. O CPM pode ser aplicado a grafos ponderados ou não ponderados.

WalkTrap

O algoritmo *WalkTrap* [Pons e Latapy, 2005] fundamenta-se na premissa de que passeios aleatórios de curtas distâncias tendem a permanecer na mesma comunidade. Como o *Fast Greedy*, o algoritmo *Walktrap* aplica uma lógica baseada no agrupamento hierárquico aglomerativo, em que nós, ou agrupamentos de nós, são recursivamente mesclados segundo um critério de união. Para lidar com a alta complexidade computacional de encontrar as comunidades ótimas, é possível empregar uma abordagem de Monte Carlo na estimação das probabilidades para os passeios aleatórios. Destaca-se que o algoritmo *Walktrap* atribui apenas uma comunidade a cada nó e, dessa forma, não há sobreposição entre as comunidades.

O algoritmo *WalkTrap* computa uma matriz de probabilidades de transição a partir de uma matriz de adjacências. Cada elemento da matriz de transição representa a probabilidade do passeio continuar para o nó adjacente com base na intensidade da relação entre os nós. Utiliza-se um processo de passeio aleatório com uma quantidade pequena

de passos para definir a probabilidade de transição entre um nó v_i e um nó v_j. Essa probabilidade é influenciada pelo grau do nó v_j de forma que existe uma maior probabilidade de transicionar para um nó com maior grau. A probabilidade de transição também é maior quando os nós v_i e v_j estão na mesma comunidade. Os passeios aleatórios definem uma distância entre os nós que é, então, generalizada para distância entre comunidades. Considera-se um passeio aleatório que se inicia em uma comunidade a partir de um nó inicial escolhido aleatoriamente e uniformemente dentre os nós da comunidade. Define-se a probabilidade de transição de partir de uma comunidade c para um nó v_j em t passos, a fim de encontrar a distância entre comunidades. Inicialmente, existem $|\mathcal{V}|$ comunidades, uma para cada nó. Computa-se todas as distâncias entre os nós adjacentes. Dados k passos, duas comunidades são escolhidas de acordo com um critério, essas duas comunidades são fundidas e as distâncias entre as comunidades atualmente existentes são computadas. Após $|\mathcal{V}|-1$ passos, o algoritmo termina. O objetivo é minimizar a média das distâncias quadradas entre cada nó e a comunidade à qual pertence. A modularidade é normalmente usada para determinar a divisão ótima entre as comunidades.

InfoMap

O algoritmo *InfoMap* foca na otimização da equação de mapa (*map equation*) [Rosvall et al., 2009], uma equação que busca minimizar o comprimento da sequência usada para representar um passeio no grafo. Essa minimização é alcançada empregando a codificação de Huffman, um tipo de codificação sem perdas que garante que os nós mais visitados sejam representados por um número menor de bits. Para minimizar o comprimento da caminhada, o grafo pode ser dividido em diferentes módulos, onde cada módulo possui seu próprio livro de códigos (*codebook*). Há também um livro de códigos que representa o movimento entre os módulos (livro de códigos de índice). O comprimento da descrição de um módulo pode ser representado pela equação do mapa:

$$L(M) = q_{\curvearrowright} H(Q) + \sum_{I=1}^{M} p^i H(P^i), \tag{17}$$

em que a primeira parte é a entropia do movimento entre as comunidades e a segunda parte é a entropia dos movimentos dentro das comunidades. As comunidades detectadas pela equação de mapa podem eventualmente divergir daquelas identificadas por algoritmos baseados na maximização da modularidade. Isso ocorre porque o cerne da equação de mapa está em otimizar o fluxo de informações, enquanto a modularidade baseia-se na conexão entre nós.

1.9.2. Métricas de Avaliação Extrínsecas e Intrínsecas

O processo de avaliação dos algoritmos de detecção de comunidade pode ser conduzido por meio de duas classes de métricas, as intrínsecas e as extrínsecas. Embora sejam igualmente úteis, a diferença entre ambas está na exigência, ou não, de uma verdade fundamental (*ground truth*), isto é, um rótulo de verdade fundamental atribuído a cada amostra. Tal rótulo funciona como base de comparação entre os resultados esperados e os obtidos. Em um cenário de detecção de câmaras de eco, esses rótulos podem expressar qual comunidade, *i.e.* câmara de eco, um usuário participa. As métricas extrínsecas

requerem obrigatoriamente a presença de amostras rotuladas, sendo portanto capazes de comparar o desempenho entre métodos. Na ausência de *ground truth*, muitos autores inferem o pertencimento, ou não, de um usuário a uma câmara de eco por meio da análise das *hashtags* compartilhadas. Apesar de subjetiva, essa inferência evita que a descoberta do rótulo de cada usuário seja dependente da análise textual com técnicas possivelmente mais demoradas e complexas de processamento de linguagem natural e aprendizado de máquina. Logo, pode-se inferir a qual comunidade um determinado usuário pertence, avaliando a ocorrência de *hashtags* em suas postagens. Além das medidas de recuperação da informação, *e.g.* acurácia, precisão, revocação, entre outras, as métricas extrínsecas também incluem:

- **Rand Index** (*RI*) que é um índice que expressa a similaridade entre os resultados previstos e os reais a partir da contabilização dos pares de amostras atribuídos na mesma ou em comunidades diferentes. Assumindo que $\mathscr{C} = \{c_1, ..., c_i\}$ é o conjunto das i comunidades retornadas pelo algoritmo de detecção de comunidade e $\mathscr{K} = \{k_1, ..., k_j\}$ é o conjunto das i comunidades de *ground truth*, o índice é dado por

$$RI(\mathscr{X}, \mathscr{Y}) = \frac{2(a+b)}{n(n-1)}, \tag{18}$$

  em que n é o número de amostras, a expressa o número de pares de amostras que mantiveram-se na mesma comunidade em $\mathscr{X}$ e $\mathscr{Y}$, e b é o número de pares de amostras que foram alocadas em comunidades diferentes em $\mathscr{X}$ e $\mathscr{Y}$. A métrica é definida entre $[0, 1]$, em que 0 indica que os dois resultados previstos e reais não concordam em nenhum par de amostras e 1 reflete a completa concordância entre os dois resultados. O Adjusted Rand Index, uma versão corrigida e simétrica da métrica original, introduz uma normalização estatisticamente induzida para produzir valores próximos de zero para partições aleatórias.

- **Informação Mútua Normalizada** (*NMI*) que é a métrica derivada da informação mútua que é calculada entre os rótulos de *ground truth* e os rótulos previstos. Para tal, considera-se que $\mathscr{X} = \{X_1, ..., X_n\}$ é o conjunto de rótulos das comunidades originalmente atribuídas às n amostras, enquanto $\mathscr{Y} = \{Y_1, ..., Y_n\}$ é o conjunto de rótulos das comunidades atribuídas às n amostras após a aplicação de um algoritmo. Ao tratar $\mathscr{X}$ e $\mathscr{Y}$ como variáveis aleatórias discretas, a informação mútua normalizada entre ambas é expressa por

$$NMI(\mathscr{X}, \mathscr{Y}) = \frac{2I(\mathscr{X}; \mathscr{Y})}{H(\mathscr{X}) + H(\mathscr{Y})}, \tag{19}$$

  em que $I(\mathscr{X} : \mathscr{Y})$ é a informação mútua entre $\mathscr{X}$ e $\mathscr{Y}$ e as entropias de $\mathscr{X}$ e $\mathscr{Y}$ são denotadas por $H(\mathscr{X})$ e $H(\mathscr{Y})$, respectivamente.

A escassez de conjuntos de dados previamente rotulados e a dificuldade de rotulagem frequentemente inviabilizam a utilização de métricas de avaliação extrínsecas. Dessa forma, cabe uma avaliação baseada em medidas intrínsecas, também chamadas de índices de validação de agrupamento, para quantificar a coesão e separação das comunidades geradas [Curiskis et al., 2020]. Dentre essas métricas, há:

- **Calinski-Harabasz Index** (*CH*) que é uma razão entre a dispersão dentro da comunidade e a dispersão entre as comunidades. Na prática, tais dispersões são obtidas através da soma de quadrados entre comunidades (SSB_c) e da soma de quadrados dentro de comunidades (SSW_c) expressos respectivamente pelas equações:

$$SSB_c = \sum_{j=1}^{|\mathscr{N}|} \left\| x_j - b_{c_j} \right\|^2 \quad (20) \qquad SSB_c = \sum_{i=1}^{|\mathscr{C}|} \|b_i - \bar{X}\|^2, \quad (21)$$

em que $|\mathscr{N}|$ é o tamanho do conjunto das amostras x_i, com $i = 1, ..., |\mathscr{N}|$ e $|\mathscr{C}|$ denota o número de comunidades. Os centróides das comunidades são denotados como b_i, em que $i = 1, ..., |\mathscr{C}|$. Adicionalmente, cada amostra x_j pertence a uma comunidade c_j cujo centroide é denotado por b_{c_j} . Diante disso, a métrica pode ser computada através da equação

$$CH = \frac{SSB_c}{SSW_c} \cdot \frac{|\mathscr{N}| - |\mathscr{K}|}{|\mathscr{K}| - 1}. \quad (22)$$

Ao contrário do índice anterior, quanto maior o valor obtido, melhor a qualidade das comunidades geradas.

- **Densidade de Partição** (D_p) que é uma particularização da métrica da densidade tradicional, que considera a compacidade das partições geradas, ou seja, das comunidades. Essa métrica é expressa por

$$D_p = \frac{2}{m} \sum_{\alpha=1}^{|\mathscr{C}|} \frac{m_\alpha - (n_\alpha - 1)}{(n_\alpha - 2), (n_\alpha - 1)}, \quad (23)$$

em que m_α e n_α são o número de arestas e vértices na comunidade $c_\alpha \in \mathscr{C}$, respectivamente.

- **Davies-Bouldin Index** (*DB*) que é um índice que fornece uma estimativa do grau de sobreposição do agrupamento. É definido como a medida de similaridade média de cada comunidade com seu par mais semelhante, em que a similaridade é a razão entre as distâncias dentro da comunidade e as distâncias entre as comunidades. Considerando $|\mathscr{C}|$ o número de comunidades, o índice é dado por

$$DB = \frac{1}{|\mathscr{C}|} \sum_{i=1}^{|\mathscr{C}|} \max_{i:j \neq i} \frac{S_i + S_j}{d_{i,j}}, \quad (24)$$

em que $S_i = \frac{1}{|c_i|} \sum_{x_j \in c_i} \left\| x_j - \upsilon_i \right\|$ é a medida de espalhamento dentro da comunidade c_i, x_j é um vetor de dimensão n atribuído à comunidade c_i, e $d_{i,j} = \left\| \upsilon_i - \upsilon_j \right\|$ é a distância euclidiana entre os centroides das comunidades c_i e c_j. Ao considerar o pior cenário de similaridade para cada comunidade, espera-se que quanto mais próximo de zero, o valor mínimo, melhor será o índice e, consequentemente, melhores serão os resultados do processo de agrupamento;

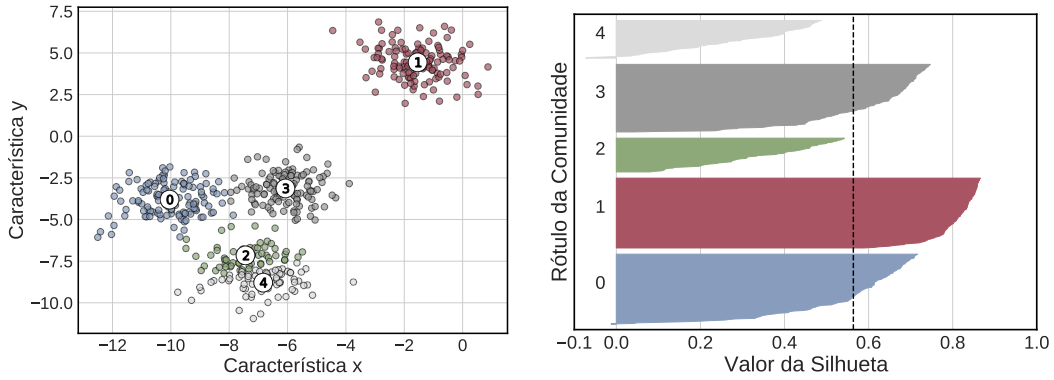
- **Coeficiente da Silhueta** (*SC*) que é um coeficiente que quantifica a qualidade do agrupamento de dados com base na proximidade e na separação entre as comunidades geradas. Conforme evidenciado na Equação 25, o coeficiente da silhueta é medido em função da média dos valores da silhueta de cada amostra do conjunto de dados. Definida entre $[-1,+1]$, a silhueta $s(i)$ para cada amostra i pode ser computada através da Equação 26.

$$SC = \frac{1}{n}\sum_{i=1}^{n} s(i) \qquad (25) \qquad\qquad s(i) = \frac{\bar{b}(i) - \bar{a}(i)}{max\left(\bar{a}(i), \bar{b}(i)\right)} \qquad (26)$$

  Na Equação 26, $\bar{a}(i)$ representa a distância média entre a amostra i em relação a todas as amostras pertencentes à mesma comunidade e $\bar{b}(i)$ reflete a distância média entre a amostra i em relação a todas as amostras. Em uma perspectiva local, o $s(i)$ avalia a adequação de cada amostra individualmente. Nesse contexto, valores próximos a 1 expressam que a amostra observada encontra-se distante das comunidades vizinhas, indicando que essa amostra foi adequadamente alocada na comunidade à qual pertence. Valores nulos indicam que a amostra localiza-se no limite ou muito próximo do limite de decisão entre duas comunidades adjacentes. Em compensação, valores negativos indicam que a amostra foi possivelmente atribuída a uma comunidade errada. Em uma perspectiva global, o *SC* traduz a qualidade do agrupamento em comunidades considerando todas o conjunto de amostras. Assim, quanto mais próximo a 1, melhor a adequação de todas as amostras às comunidades a que pertencem. De maneira oposta, valores negativos informam que o processo de agrupamento não foi adequado. A Figura 1.11 mostra graficamente os reflexos no valor da silhueta quando adotando uma divisão em comunidades não ótima.

1.10. Ferramentas de Caracterização de Câmaras de Eco

Embora seja desafiadora, a análise de câmaras de eco suscita diversas oportunidades de pesquisa devido à prevalência dessas estruturas nas redes sociais em várias plataformas. Estimativas apontam que a fonte dos desafios decorrentes das câmaras de eco é o fato de que as câmaras de eco têm muitos participantes distintos: i) os membros da câmara de eco, ii) as plataformas de mídia social e iii) o mundo "*offline*". Cada um desses participantes apresenta desafios e problemas em aberto para serem resolvidos. A existência do elemento humano nas câmaras de eco tornam o estudo dessas estruturas desafiador. Trabalhos relacionados à câmaras de eco e polarização devem considerar como as pessoas dentro da câmara de eco consomem conteúdo, percebem o mundo e veem pessoas externas à câmara de eco. Os membros da câmara de eco têm quatro características críticas que contribuem para dificultar a análise dessas estruturas: i) eles não estão cientes de que fazem parte da câmara de eco; ii) eles selecionam apenas conteúdos que aderem às suas crenças; iii) resistem a qualquer informação que refute suas crenças; iv) desconfiam de qualquer ajuda que venha de fora da sua câmara de eco. O aumento da conscientização sobre as câmaras de eco e seus efeitos no indivíduo e na sociedade é um passo essencial em direção à dissolução de câmaras de eco nocivas. Uma vez liberto de suas câmaras de eco, o indivíduo pode tornar mais civilizada e diversa sua participação no ambiente *online* [Alatawi et al., 2021].



(a) Visualização Espacial das Amostras.

(b) Silhueta das Comunidades.

Figura 1.11. Valores da silhueta para um conjunto de dados dividido em cinco comunidades. Em 1.11(a) percebe-se que as comunidades 2 e 4 estão muito próximas entre si, fato que contribui para o baixo valor da métrica nessas comunidades. Em 1.11(b) mostra-se que cada silhueta, *i.e.*, manchas coloridas, tem comprimento horizontal e espessura diretamente proporcionais ao valor da silhueta das amostras associadas e à quantidade de amostras na comunidade, respectivamente. Em um cenário ideal de agrupamento em comunidades, o valor da silhueta correspondente a cada comunidade deve ser maior que o valor médio dos coeficientes de silhueta (linha pontilhada preta). Além disso, espera-se que nesse cenário as espessuras de cada silhueta sejam semelhantes entre si.

Desenvolvida por Gillani *et al.*, a **Social Mirror**[14] é um aplicação *web* que permite aos usuários explorarem, de forma interativa, suas conexões politicamente ativas no *Twitter* [Gillani et al., 2018]. Tais conexões são visualmente modeladas na forma de um grafo, em que os nós representam um conjunto de contas participantes de um debate específico e as arestas representam uma relação de amizade mútua entre as contas. Internamente, a *Social Mirror* emprega o *PageRank* e um classificador de ideologia política para embasar o dimensionamento dos nós e atribuir uma tonalidade a eles. Ao apresentar uma visão panorâmica da fração mais ideologicamente fragmentada da rede do usuário, a ferramenta visa inspirar a autorreflexão e motivar o compartilhamento de conteúdo mais diversificado entre usuários.

O **ChamberBreaker** [Jeon et al., 2021] é um sistema baseado em jogos, projetado para aumentar as capacidades cognitivas dos jogadores a fim de torná-los mais aptos a responderem preventivamente ao surgimento de câmaras de eco. Dentro do *ChamberBreaker*, cada jogador torna-se um usuário anônimo da rede social e é induzido a compartilhar continuamente *tweets* tendenciosos que fomentam a criação de uma câmara de eco. O sistema integra conceitos psicológicos e disponibiliza diferentes temas, cada um abordando uma característica específica das câmaras de eco. Dentre os benefícios do sistema aos seus usuários pode-se citar o aumento das intenções pelo consumo de informações com perspectivas mais diversas, bem como a elevação da conscientização sobre os efeitos negativos do fenômeno da câmara de eco.

Focando na redução do consumo de mídia partidária, **Balancer** [Munson et al.,

[14]Disponível em https://socialmirror.media.mit.edu.

2013] é uma aplicação curta para navegador (*widget*) que exibe aos usuários se seu histórico de leitura é consistente com um padrão de leitura ideologicamente balanceado. Ao medir continuamente essa inclinação, prova-se que a ferramenta promove melhorias, mesmo que pequenas, nos hábitos de leitura enviesados politicamente. Em um estudo de campo, usuários relatam mais visitas a páginas *web* ideologicamente opostos e centristas.

Para quantificar a força da câmara de eco à qual o usuário pertence, o **Check-my-echo** [Bail et al., 2018] analisa previamente a orientação política proeminente das contas que o usuário segue, sejam elas de políticos eleitos, jornalistas ou grupos de defesa de temas específicos. Tal análise aplica um algoritmo de particionamento recursivo para encontrar padrões de conexão entre líderes de opinião e autoridades eleitas de diferentes inclinações políticas. Assim, ao efetuar o *login* no *Twitter* através da ferramenta, o usuário consegue descobrir sua pontuação média na escala de 0 a 10 de inclinação política. Valores próximos ao limite inferior indicam um caráter mais liberal, enquanto que valores próximo ao limite superior refletem um cunho mais conservador. A assertividade da ferramenta é maior para usuários dos Estados Unidos, visto que os líderes de opinião usados como base de comparação são do cenário político estadunidense.

Ao investigar o surgimento das câmaras de eco, Sasahara *et al.* desenvolvem um modelo simples de compartilhamento de informações em redes sociais *online* [Sasahara et al., 2021]. Na prática, o modelo retrata o comportamento do usuário quando continuamente exposto a mensagens semelhantes endossadas por amigos. Para facilitar a exploração do modelo, os autores desenvolvem uma demonstração interativa[15] que permite executar simulações ajustando três parâmetros-chave: i) a tolerância, relacionada a como o usuário lida com diferentes opiniões; ii) a influência social, que controla o quão rápido a opinião do usuário pode ser atualizada; e iii) o *unfriending*, que regula a frequência do ato de desfazer amizades. A demonstração evidencia que o processo de evolução das câmaras de eco atinge um estado estacionário, caracterizado por dois aspectos distintos, a polarização de opinião e a segregação da rede.

1.11. Atividade Prática

Esta seção consolida a pluralidade de conceitos teóricos abordados no capítulo através de uma atividade prática do processo de estruturação, caracterização e detecção de câmaras de eco em redes sociais. O processo ocorre na rede social *Twitter* e é representado na Figura 1.11. O processo é totalmente desenvolvido na linguagem *Python* e inclui quatro etapas contendo seis tarefas: i) coleta de informações textuais relacionadas a usuários potencialmente integrantes de câmaras de eco, empregando a interface de programação de aplicação (*Application Programming Interface* - API) do *Twitter* para efetuar o *web scraping*; ii) estruturação de grafos relacionais com base nos dados coletados dos usuários; iii) caracterização do grafo gerado segundo métricas de redes complexas; iv) aplicação eficiente de algoritmos de detecção de comunidade; v) execução de um método de incorporação de grafos a fim de gerar uma representação vetorial densa do grafo criado; e vi) avaliação da eficiência e qualidade da detecção, tendo como parâmetros os índices de validação de agrupamento.

A primeira etapa da atividade prática consiste em exemplificar a tarefa de cons-

[15]Disponível em https://osome.iu.edu/demos/echo/.

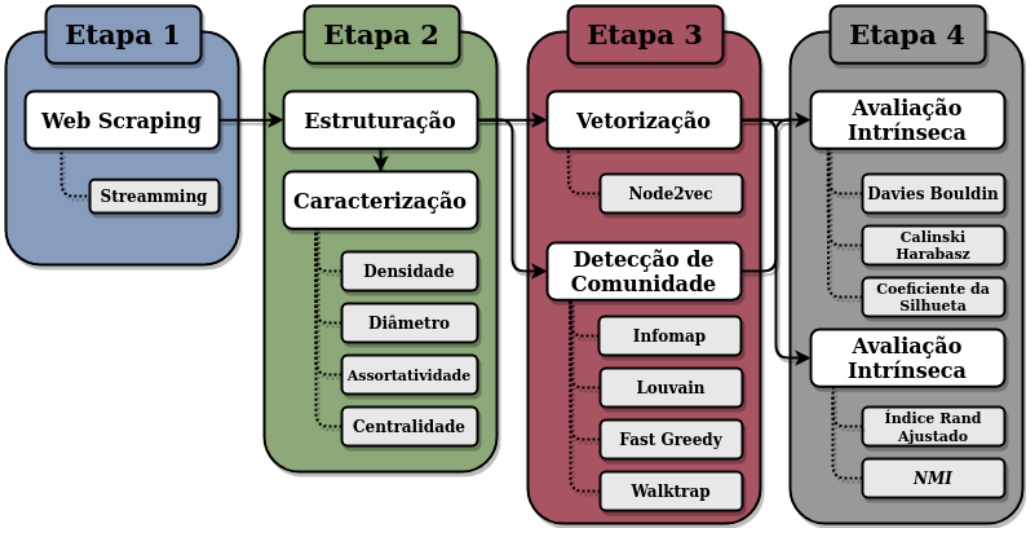


Figura 1.12. Fluxograma do processo de detecção de câmaras de eco desenvolvido na atividade prática. A primeira etapa compreende a coleta de dados no *Twitter* para construir uma base de dados. A segunda etapa prevê a estruturação dos dados em um grafo relacional para ser caracterizado por métricas de redes complexas. Na terceira etapa, o grafo é submetido a uma técnica de incorporação e a diferentes algoritmos de detecção de comunidade. A quarta etapa avalia a qualidade da detecção através de métricas intrínsecas e extrínsecas.

trução de uma base de dados relacionada à câmaras de eco, a partir do *web scraping* do *Twitter*. Contudo, como descrito na Seção 1.5, a estratégia de construção mais difundida na literatura é através da captura de postagens sobre tópicos controversos. Além de favorecer a polarização de opiniões, um tópico controverso costuma ser intensamente debatido nas redes sociais, gerando um alto engajamento e consequentemente dados para coleta. Perante essas observações, a atividade prática opta pelo uso de uma palavra-chave para nortear o processo de aquisição de dados. Para automatizar o processo de obtenção dos *tweets* relacionados à palavra-chave selecionada, adota-se um *script*[16] em *Python* que acessa a API do *Twitter* usando credenciais de desenvolvedor. Em posse dessas credenciais, a biblioteca **Tweepy**[17] permite a extração de conteúdo textual dos *tweets* de qualquer perfil aberto na rede social segundo duas abordagens. A primeira abordagem destina-se à captura de metadados históricos, postados até o momento de execução do *script*. Apesar da simplicidade e do imediatismo da coleta, a abordagem enfrenta duas limitações impostas pelo *Twitter*, uma temporal e outra quantitativa. A limitação temporal remete a um número máximo de requisições passíveis de serem direcionadas à plataforma a cada janela de tempo de 15 minutos. Paralelamente, há uma limitação na quantidade de *tweets* históricos passíveis de serem coletados. Dependendo da permissão vinculada às credenciais, a obtenção de *tweets* é restringida a um período de até, no máximo, alguns meses passados a contar pela data de execução do *script*. Ciente dessas limitações e visando a dinamicidade e contemporaneidade da captura, a atividade prática foca na segunda abordagem de coleta. Diferentemente da anterior, essa abordagem prevê a captura ininterrupta do fluxo de *tweets* relacionado à palavra-chave, publicados a partir do momento de execução

[16]Disponível em https://github.com/nicollasro/Echochamber.
[17]Disponível em https://www.tweepy.org/.

do *script*. Logo, quaisquer *tweets* futuros publicados contendo a palavra-chave adotada, serão coletados.

Diante da ampla variedade de campos retornáveis pela coleta em fluxo contínuo de *tweets*, é recomendado que estes dados sejam adequadamente organizados para análises subsequentes. Para esse propósito, utiliza-se o **Pandas**[18], uma biblioteca capaz de prover a manipulação e alocação em memória de dados a partir de duas estruturas de dados primárias: `Series`, indicada para conjuntos de dados unidirecionais; e `Dataframes`, destinada a conjuntos de dados bidimensionais. Embora a atividade prática exemplifique a tarefa de coleta de dados utilizando o *web scraping*, a formação de uma base de dados contundente pode demorar algumas horas dependendo do tópico e da estabilidade da conexão. Por esta razão, opta-se pela utilização de uma base de dados previamente processada e correlacionada. Essa base de dados[19] foi originalmente consolidada por Morini *et al.* e contém três conjuntos de dados relacionados a câmaras de eco [Morini et al., 2021]. Cada conjunto foi construído monitorando uma questão sociopolítica durante cinco semestres consecutivos[20], resultando assim em cinco *snapshots* semestrais do período total.

A segunda etapa da atividade prática compreende a estruturação dos dados em grafos relacionais seguida da caracterização dos mesmos. Utiliza-se um segundo *script* em *Python* para realizar as tarefas necessárias. Em *Python*, a criação de estruturas em grafo é facilmente implementável por funções da bibliotecas **igraph**[21] ou **NetworkX**[22]. Por ser implementado na linguagem C, o *igraph* detém um desempenho consideravelmente superior ao de bibliotecas puramente desenvolvidas em Python, como é o caso da *NetworkX*. Além disso, o *igraph* dispõe de uma gama maior de algoritmos de detecção de comunidade nativamente implementados. Em compensação, a *NetworkX* possui uma documentação mais detalhada e uma comunidade *online* mais ativa. Devido a esses fatores, a atividade prática emprega ambas as bibliotecas de forma complementar, destinando à *NetworkX* as tarefas de estruturação e caracterização, enquanto a *igraph* é usada na tarefa de detecção de comunidades.

Dentre as formas de construir um grafo na *NetworkX*, a mais comum inicia com a criação do objeto referente ao grafo a partir de uma classe. A fim de reproduzir uma rede que expresse as relações de amizade entre usuários, por exemplo, pode-se utilizar a classe `MultiDiGraph`. Essa classe permite instanciar um grafo direcionado que represente as relações de seguidor (*follower*) e seguindo (*following*) entre usuários. Na prática, a inclusão de nós e suas respectivas arestas ponderadas pode ser alcançada através das funções `add_nodes` e `add_weighted_edges_from`. Embora utilizem classes distintas (`Graph`, `DiGraph` ou `MultiGraph`), a mesma lógica aplica-se na criação de grafos de *retweets* e nos grafos de menções entre usuários. Uma vez estruturado, o grafo gerado pode ser submetido a uma caracterização estrutural segundo várias métricas de redes complexas. Através de funções como `diameter` e `density`, pode-se mensurar o diâmetro e a densidade. Adicionalmente, a função `degree_assortativity_coefficient`

[18]Disponível em https://pandas.pydata.org/.
[19]Disponível em https://github.com/virgiiim/EC_Reddit_CaseStudy.
[20]Detalhes estruturais sobre o conjunto de dados estão expressos na Tabela 1.3.
[21]Disponível em https://igraph.org/python/.
[22]Disponível em http://networkx.lanl.gov/.

permite medir a assortatividade do grafo considerando o grau dos nós como característica.

Um dos objetivos da terceira etapa é demonstrar a tarefa de vetorização utilizando incorporação de grafos. Essa tarefa é executado pelo **PecanPy**[23], uma implementação do *Node2vec* rápida, paralelizada, eficiente em memória e otimizada para usar memória cache [Liu e Krishnan, 2021]. O PecanPy opera principalmente em três modos distintos, cada um otimizado para redes com características de tamanho e densidade específicas. Em especial, o modo `PreComp` destina-se a redes pequenas, ou seja, àquelas contendo menos que $10k$ nós, independente da densidade. Essa limitação estrutural é derivada da computação e armazenamento antecipados de todas as probabilidades de transição de segunda ordem do grafo original. O modo `SparseOTF` é indicado para redes grandes e esparsas, compostas preferencialmente por um número de nós inferior a $10k$ e possuindo até 10% de arestas possíveis. Em contrapartida, o modo `DenseOTF` foca no processamento de redes grandes e densas, aquelas contendo mais de $10k$ nós e mais que 10% das arestas possíveis. Para lidar com redes de grandes proporções, ambos os modos calculam as probabilidades de transição de segunda ordem *On-The-Fly* (OTF) durante a geração de passeios aleatórios, sem salvá-las. Além da redução do custo de memória, essa otimização evita o desperdício computacional atrelado ao cálculo de probabilidades de transição de segunda ordem que eventualmente nunca seriam utilizadas na geração de passeios. Além da seleção do modo de operação, o Pecanpy permite configurar o tamanho do vetor incorporado de saída.

A terceira etapa também demonstra a tarefa de execução de alguns algoritmos de detecção de comunidade, sendo eles *Infomap*, *Louvain*, *Fast Greedy* e *Walktrap*. A escolha dos algoritmos tem a finalidade de diversificar os resultados gerados, uma vez que cada algoritmo utiliza uma métrica base ou lógica específica de funcionamento. Embora admitam a configuração de outros parâmetros opcionais, todos esses algoritmos requerem obrigatoriamente um grafo previamente estruturado como parâmetro de entrada. Como saída, cada algoritmo retorna o rótulo da comunidade atribuído a cada nó. Após obter as saídas de ambas as tarefas de vetorização e detecção de comunidades, segue-se para a etapa de avaliação.

Na quarta e última etapa da atividade prática, diversas funções e classes de módulos específicos da biblioteca **Scikit-learn**[24] são empregadas na avaliação das comunidades, *i.e.* câmaras de eco, detectadas. Ao incluir algumas funções do módulo `Metrics` ao *script* desenvolvido, pode-se implementar métricas intrínsecas e extrínsecas de avaliação. Em especial, funções como `calinski_harabasz_score`, `davies_bouldin_score` e `silhouette_score` permitem avaliar o quão compactas e sobrepostas são as comunidades geradas, retornando os valores dos índices Calinski-Harabasz e Davies-Bouldin e do Coeficiente da Silhueta. Aproveitando a existência de um conjunto de dados pré-rotulado, pode-se também avaliar extrinsecamente os resultados através de funções como `adjusted_rand_score` e `normalized_mutual_info_score`. Tais funções retornam respectivamente os valores do índice Rand Ajustado e da Informação Mútua Normalizada entre os rótulos obtidos e os de referência. Ao final da atividade, evidenciam-se os diferentes resultados obtidos pelos algoritmos de detecção de comunidade adotados, tanto numa perspectiva

[23]Disponível em https://github.com/krishnanlab/PecanPy.

[24]Disponível em https://scikit-learn.org/stable/.

quantitativa quanto visual. A visualização dos resultados é realizada por meio de funções da biblioteca **Matplotlib**[25] capaz de exibir gráficos em múltiplos formatos.

1.12. Considerações Finais

Este minicurso investiga as câmaras de eco (*echo chamber*), um fenômeno relacionado às estruturas sociais homogêneas cujos membros excluem sistematicamente opiniões, crenças e fontes de informação que divirjam daquelas disseminadas entre eles. Para mitigar equívocos semânticos a respeito do tema, busca-se inicialmente descrever as definições e características das câmaras de eco e dos principais termos correlatos. O minicurso pauta-se na metodologia PRISMA para reunir 30 publicações relacionadas à detecção e caracterização de câmaras de eco. Tais trabalhos compõem a base do conteúdo técnico detalhado ao longo do minicurso. Dentre as duas principais abordagens de identificação de câmaras de eco em redes sociais, o minicurso adentra nos procedimentos e técnicas empregadas na abordagem topológica. A abordagem topológica visa detectar câmaras de eco segundo um viés estrutural, o qual busca padrões de conexão característicos de usuários pertencentes a essas estruturas ressonantes de informação. Durante a descrição das principais etapas no processo de detecção, o minicurso inicialmente relata a construção de uma base de dados relacionada a câmaras de eco a partir da coleta automática de postagens sobre um tópico controverso. Após a coleta, informações específicas podem ser extraídas e correlacionadas em uma estrutura em grafo. Essa estruturação permite que a tarefa de identificação de câmaras de eco seja interpretada como um problema de descoberta de comunidades em grafos. Como consequência direta dessa associação, pode-se caracterizar estruturalmente as câmaras de eco empregando tanto métricas genéricas de redes complexas, quanto métricas específicas como a controvérsia. Sendo o cerne do processo de identificação de câmaras de eco, os algoritmos de detecção de comunidade são vastamente explorados no minicurso, em que suas lógicas de funcionamento, aplicabilidade e complexidade são descritas. Paralelamente, o minicurso debate os principais modelos de vetorização com ênfase nas técnicas de incorporação de grafos. Ao abordar diferentes cenários de avaliação, o minicurso inclui métricas extrínsecas que são dependentes de uma verdade fundamental (*ground truth*), assim como métricas intrínsecas que são independentes desse rótulo de referência. Ambientada no *Twitter*, a atividade prática desenvolvida mostra em detalhes a viabilidade da detecção de câmaras de eco utilizando técnicas computacionais. Diante disso, o minicurso apresenta diversos conceitos sociais e técnicos que permeiam as câmaras de eco no âmbito de redes sociais *online*. Assim, este minicurso fomenta a análise crítica e motiva pesquisadores a desenvolver soluções que auxiliem na identificação e prevenção de câmaras de eco em redes sociais, mitigando assim os seus efeitos danosos. Vale ressaltar que promover a interação saudável entre usuários de redes sociais *online* é uma responsabilidade conjunta da comunidade científica, formuladores de políticas digitais, administração e da sociedade em geral.

Referências

[Ahmed et al., 2013] Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V. e Smola, A. J. (2013). Distributed large-scale natural graph factorization. Em *Proceedings of the 22nd International Conference on World Wide Web*, p. 37–48.

[25] Disponível em https://matplotlib.org/.

[Alatawi et al., 2021] Alatawi, F., Cheng, L., Tahir, A., Karami, M., Jiang, B., Black, T. e Liu, H. (2021). A survey on echo chambers on social media: Description, detection and mitigation. *arXiv preprint arXiv:2112.05084*.

[Alduaiji et al., 2018] Alduaiji, N., Datta, A. e Li, J. (2018). Influence propagation model for clique-based community detection in social networks. *IEEE Transactions on Computational Social Systems*, 5(2):563–575.

[Alsini et al., 2020] Alsini, A., Datta, A. e Huynh, D. Q. (2020). On utilizing communities detected from social networks in hashtag recommendation. *IEEE Transactions on Computational Social Systems*, 7(4):971–982.

[Bail et al., 2018] Bail, C. A., Argyle, L. P., Brown, T. W., Bumpus, J. P., Chen, H., Hunzaker, M. F., Lee, J., Mann, M., Merhout, F. e Volfovsky, A. (2018). Exposure to opposing views on social media can increase political polarization. *Proceedings of the National Academy of Sciences*, 115(37):9216–9221.

[Bakshy et al., 2015] Bakshy, E., Messing, S. e Adamic, L. A. (2015). Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132.

[Barabási, 2016] Barabási, A.-L. (2016). *Network Science*. "Cambridge University Press".

[Barberá et al., 2015] Barberá, P., Jost, J. T., Nagler, J., Tucker, J. A. e Bonneau, R. (2015). Tweeting from left to right: Is online political communication more than an echo chamber? *Psychological science*, 26(10):1531–1542.

[Baumann et al., 2020] Baumann, F., Lorenz-Spreen, P., Sokolov, I. M. e Starnini, M. (2020). Modeling echo chambers and polarization dynamics in social networks. *Phys. Rev. Lett.*, 124:048301.

[Belkin e Niyogi, 2001] Belkin, M. e Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*, 14.

[Bessi, 2016] Bessi, A. (2016). Personality traits and echo chambers on facebook. *Computers in Human Behavior*, 65:319–324.

[Bessi et al., 2015] Bessi, A., Zollo, F., Del Vicario, M., Scala, A., Caldarelli, G. e Quattrociocchi, W. (2015). Trend of narratives in the age of misinformation. *PloS one*, 10(8):e0134641.

[Cao et al., 2015] Cao, S., Lu, W. e Xu, Q. (2015). Grarep: Learning graph representations with global structural information. Em *Proceedings of the 24th ACM International on Conference on Information and Knowledge management*, p. 891–900.

[Cao et al., 2016] Cao, S., Lu, W. e Xu, Q. (2016). Deep neural networks for learning graph representations. Em *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

[Chitra e Musco, 2020] Chitra, U. e Musco, C. (2020). Analyzing the impact of filter bubbles on social network polarization. Em *Proceedings of the 13th International Conference on Web Search and Data Mining*, p. 115–123.

[Cinelli et al., 2021] Cinelli, M., De Francisci Morales, G., Galeazzi, A., Quattrociocchi, W. e Starnini, M. (2021). The echo chamber effect on social media. *Proceedings of the National Academy of Sciences*, 118(9):e2023301118.

[Cinus et al., 2022] Cinus, F., Minici, M., Monti, C. e Bonchi, F. (2022). The effect of people recommenders on echo chambers and polarization. Em *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, p. 90–101.

[Clauset et al., 2004] Clauset, A., Newman, M. E. e Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6):066111.

[Colleoni et al., 2014] Colleoni, E., Rozza, A. e Arvidsson, A. (2014). Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of communication*, 64(2):317–332.

[Conover et al., 2011] Conover, M., Ratkiewicz, J., Francisco, M., Gonçalves, B., Menczer, F. e Flammini, A. (2011). Political polarization on twitter. Em *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, p. 89–96.

[Cossard et al., 2020] Cossard, A., De Francisci Morales, G., Kalimeri, K., Mejova, Y., Paolotti, D. e Starnini, M. (2020). Falling into the echo chamber: The italian vaccination debate on twitter. Em *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, p. 130–140.

[Cota et al., 2019] Cota, W., Ferreira, S. C., Pastor-Satorras, R. e Starnini, M. (2019). Quantifying echo chamber effects in information spreading over political communication networks. *EPJ Data Science*, 8(1):1–13.

[Cui et al., 2019] Cui, P., Wang, X., Pei, J. e Zhu, W. (2019). A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):833–852.

[Curiskis et al., 2020] Curiskis, S. A., Drake, B., Osborn, T. R. e Kennedy, P. J. (2020). An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit. *Information Processing & Management*, 57(2):102034.

[Dash et al., 2019] Dash, A., Mukherjee, A. e Ghosh, S. (2019). A network-centric framework for auditing recommendation systems. Em *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, p. 1990–1998.

[de Oliveira et al., 2020a] de Oliveira, N. R., Medeiros, D. S. e Mattos, D. M. (2020a). A sensitive stylistic approach to identify fake news on social networking. *IEEE Signal Processing Letters*, 27:1250–1254.

[de Oliveira et al., 2021a] de Oliveira, N. R., Medeiros, D. S. e Mattos, D. M. (2021a). Caracterização sócio-temporal de conteúdos em redes sociais baseada em processamento em fluxo. Em *Anais do XXVI Workshop de Gerência e Operação de Redes e Serviços*, p. 54–67. SBC.

[de Oliveira et al., 2020b] de Oliveira, N. R., Pisa, P. S., Costa, B., Lopez, M. A., Moraes, I. M. e Mattos, D. M. (2020b). Processamento de linguagem natural para identificação de notícias falsas em redes sociais: Ferramentas, tendências e desafios. Em *Minicursos do XX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg 2020)*.

[de Oliveira et al., 2021b] de Oliveira, N. R., Pisa, P. S., Lopez, M. A., de Medeiros, D. S. V. e Mattos, D. M. F. (2021b). Identifying fake news on social networks based on natural language processing: Trends and challenges. *Information*, 12(1).

[Deffuant et al., 2000] Deffuant, G., Neau, D., Amblard, F. e Weisbuch, G. (2000). Mixing beliefs among interacting agents. *Advances in Complex Systems*, 3(01n04):87–98.

[Del Vicario et al., 2016] Del Vicario, M., Vivaldo, G., Bessi, A., Zollo, F., Scala, A., Caldarelli, G. e Quattrociocchi, W. (2016). Echo chambers: Emotional contagion and group polarization on facebook. *Scientific reports*, 6(1):1–12.

[Donkers e Ziegler, 2021] Donkers, T. e Ziegler, J. (2021). The dual echo chamber: Modeling social media polarization for interventional recommending. Em *Fifteenth ACM Conference on Recommender Systems*, RecSys '21, p. 12–22, New York, NY, USA. Association for Computing Machinery.

[Fletcher et al., 2021] Fletcher, R., Robertson, C. T. e Nielsen, R. K. (2021). How many people live in politically partisan online news echo chambers in different countries? *Journal of Quantitative Description: Digital Media*, 1.

[Garimella et al., 2017] Garimella, K., De Francisci Morales, G., Gionis, A. e Mathioudakis, M. (2017). Reducing controversy by connecting opposing views. Em *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, p. 81–90.

[Garimella et al., 2018a] Garimella, K., De Francisci Morales, G., Gionis, A. e Mathioudakis, M. (2018a). Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. Em *Proceedings of the 2018 World Wide Web Conference - WWW '18*, p. 913–922.

[Garimella et al., 2018b] Garimella, K., Morales, G. D. F., Gionis, A. e Mathioudakis, M. (2018b). Quantifying controversy on social media. *Trans. Soc. Comput.*, 1(1).

[Garimella e Weber, 2017] Garimella, V. R. K. e Weber, I. (2017). A long-term analysis of polarization on twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):528–531.

[Gausen et al., 2022] Gausen, A., Luk, W. e Guo, C. (2022). Using agent-based modelling to evaluate the impact of algorithmic curation on social media. *ACM Journal of Data and Information Quality (JDIQ)*.

[Ge et al., 2020] Ge, Y., Zhao, S., Zhou, H., Pei, C., Sun, F., Ou, W. e Zhang, Y. (2020). Understanding echo chambers in e-commerce recommender systems. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 2261–2270.

[Gillani et al., 2018] Gillani, N., Yuan, A., Saveski, M., Vosoughi, S. e Roy, D. (2018). Me, my echo chamber, and i: introspection on social media polarization. Em *Proceedings of the 2018 World Wide Web Conference*, p. 823–831.

[Goyal e Ferrara, 2018] Goyal, P. e Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94.

[Grover e Leskovec, 2016] Grover, A. e Leskovec, J. (2016). node2vec: Scalable feature learning for networks. Em *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 855–864.

[Guerra et al., 2013] Guerra, P., Meira Jr, W., Cardie, C. e Kleinberg, R. (2013). A measure of polarization on social media networksbased on community boundaries. Em *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013*, p. 215–224.

[Jeon et al., 2021] Jeon, Y., Kim, B., Xiong, A., Lee, D. e Han, K. (2021). Chamberbreaker: Mitigating the echo chamber effect and supporting information hygiene through a gamified inoculation system. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–26.

[Liu e Krishnan, 2021] Liu, R. e Krishnan, A. (2021). Pecanpy: a fast, efficient and parallelized python implementation of node2vec. *Bioinformatics*, 37(19):3377–3379.

[Meyer-Baese e Schmid, 2014] Meyer-Baese, A. e Schmid, V. (2014). Chapter 2 - feature selection and extraction. Em Meyer-Baese, A. e Schmid, V., editors, *Pattern Recognition and Signal Analysis in Medical Imaging (Second Edition)*, p. 21–69. Academic Press, Oxford, second edition edição.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G. e Dean, J. (2013). Efficient estimation of word representations in vector space. Em *1st International Conference on Learning Representations, ICLR*, p. 1–12.

[Minici et al., 2022] Minici, M., Cinus, F., Monti, C., Bonchi, F. e Manco, G. (2022). Cascade-based echo chamber detection. *arXiv preprint arXiv:2208.04620*.

[Morales et al., 2021] Morales, G. D. F., Monti, C. e Starnini, M. (2021). No echo in the chambers of political interactions on reddit. *Scientific reports*, 11(1):1–12.

[Morini et al., 2021] Morini, V., Pollacci, L. e Rossetti, G. (2021). Toward a standard approach for echo chamber detection: Reddit case study. *Applied Sciences*, 11(12).

[Munson et al., 2013] Munson, S., Lee, S. e Resnick, P. (2013). Encouraging reading of diverse political viewpoints with a browser widget. Em *Proceedings of The International AAAI Conference on Web and Social Media*, volume 7, p. 419–428.

[Nguyen, 2020] Nguyen, C. T. (2020). Echo chambers and epistemic bubbles. *Episteme*, 17(2):141–161.

[Ou et al., 2016] Ou, M., Cui, P., Pei, J., Zhang, Z. e Zhu, W. (2016). Asymmetric transitivity preserving graph embedding. Em *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1105–1114.

[Page et al., 1999] Page, L., Brin, S., Motwani, R. e Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Relatório técnico, Stanford InfoLab.

[Perozzi et al., 2014] Perozzi, B., Al-Rfou, R. e Skiena, S. (2014). Deepwalk: Online learning of social representations. Em *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 701–710.

[Pons e Latapy, 2005] Pons, P. e Latapy, M. (2005). Computing communities in large networks using random walks. Em *International symposium on computer and information sciences*, p. 284–293. Springer.

[Rossetti e Cazabet, 2018] Rossetti, G. e Cazabet, R. (2018). Community discovery in dynamic networks: A survey. *ACM Comput. Surv.*, 51(2).

[Rosvall et al., 2009] Rosvall, M., Axelsson, D. e Bergstrom, C. T. (2009). The map equation. *The European Physical Journal Special Topics*, 178(1):13–23.

[Roweis e Saul, 2000] Roweis, S. T. e Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.

[Sasahara et al., 2021] Sasahara, K., Chen, W., Peng, H., Ciampaglia, G. L., Flammini, A. e Menczer, F. (2021). Social influence and unfollowing accelerate the emergence of echo chambers. *Journal of Computational Social Science*, 4(1):381–402.

[Sîrbu et al., 2019] Sîrbu, A., Pedreschi, D., Giannotti, F. e Kertész, J. (2019). Algorithmic bias amplifies opinion fragmentation and polarization: A bounded confidence model. *PloS one*, 14(3):e0213246.

[Tang et al., 2015] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J. e Mei, Q. (2015). Line: Large-scale information network embedding. Em *Proceedings of the 24th international conference on world wide web*, p. 1067–1077.

[Terren e Borge-Bravo, 2021] Terren, L. e Borge-Bravo, R. (2021). Echo chambers on social media: a systematic review of the literature. *Review of Communication Research*, 9:99–118.

[Tokita et al., 2021] Tokita, C. K., Guess, A. M. e Tarnita, C. E. (2021). Polarized information ecosystems can reorganize social networks via information cascades. *Proceedings of the National Academy of Sciences*, 118(50):e2102147118.

[Törnberg, 2018] Törnberg, P. (2018). Echo chambers and viral misinformation: Modeling fake news as complex contagion. *PLoS one*, 13(9):e0203958.

[Villa et al., 2021] Villa, G., Pasi, G. e Viviani, M. (2021). Echo chamber detection and analysis. *Social Network Analysis and Mining*, 11(1):1–17.

[Wandelt et al., 2020] Wandelt, S., Shi, X. e Sun, X. (2020). Complex network metrics: Can deep learning keep up with tailor-made reference algorithms? *IEEE Access*, 8:68114–68123.

[Wang et al., 2016] Wang, D., Cui, P. e Zhu, W. (2016). Structural deep network embedding. Em *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 1225–1234.

[Welling e Kipf, 2016] Welling, M. e Kipf, T. N. (2016). Semi-supervised classification with graph convolutional networks. Em *J. International Conference on Learning Representations (ICLR 2017)*.

[Williams et al., 2015] Williams, H. T., McMurray, J. R., Kurz, T. e Lambert, F. H. (2015). Network analysis reveals open forums and echo chambers in social media discussions of climate change. *Global environmental change*, 32:126–138.

[Xu, 2021] Xu, M. (2021). Understanding graph embedding methods and their applications. *SIAM Review*, 63(4):825–853.

[Yang et al., 2016] Yang, Z., Algesheimer, R. e Tessone, C. J. (2016). A comparative analysis of community detection algorithms on artificial networks. *Scientific reports*, 6(1):1–18.

[Yue et al., 2020] Yue, X., Wang, Z., Huang, J., Parthasarathy, S., Moosavinasab, S., Huang, Y., Lin, S. M., Zhang, W., Zhang, P. e Sun, H. (2020). Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics*, 36(4):1241–1251.

[Zannettou et al., 2018] Zannettou, S., Bradlyn, B., De Cristofaro, E., Kwak, H., Sirivianos, M., Stringini, G. e Blackburn, J. (2018). What is gab: A bastion of free speech or an alt-right echo chamber. Em *Companion Proceedings of the The Web Conference 2018*, p. 1007–1014.

[Zollo et al., 2017] Zollo, F., Bessi, A., Del Vicario, M., Scala, A., Caldarelli, G., Shekhtman, L., Havlin, S. e Quattrociocchi, W. (2017). Debunking in a world of tribes. *PloS one*, 12(7):e0181821.

Capítulo 2

Processamento de Linguagem Natural em Textos de Mídias Sociais: Fundamentos, Ferramentas e Aplicações

Frances A. Santos[1], Jordan K. Kobellarz[2], Fábio R. de Souza[3], Leandro A. Villas[1] e Thiago H. Silva[2]

[1]Instituto de Computação, Unicamp, Campinas, Brasil

[2]Departamento Acadêmico de Informática, UTFPR, Curitiba, Brasil

[3]Faculdade de Filosofia, Letras e Ciências Humanas, USP, São Paulo, Brasil

{frances.santos,leandro}@ic.unicamp.br, thiagoh@utfpr.edu.br,

jordan@alunos.utfpr.edu.br, fabiorezende@usp.br

Abstract

Social media have proved to be an essential data source for analyzing social phenomena and their impacts in the real world due to the engagement and plurality of its users and the content shared by them. As the majority of content shared is textual, and we can find massive volumes of data, natural language processing techniques, jointly with machine learning models, are crucial to monitor the manifestation and evolution of these phenomena over time. This chapter presents an overview of the entire process of analyzing social media texts by computational means, from social media data extraction and main textual analysis techniques to examples of applications. Although this chapter focuses on social media texts and their particularities, most of the techniques presented can also be applied to other textual sources.

Resumo

As mídias sociais têm se mostrado uma importante fonte de dados para análise de fenômenos sociais e seus impactos no mundo real, devido ao engajamento e pluralidade de seus usuários e dos conteúdos compartilhados por eles. Como boa parte do conteúdo dessas fontes é textual e se tratando de um volume massivo de dados, torna-se fundamental o emprego de técnicas de processamento de linguagem natural e sua combinação com modelos de aprendizado de máquina para análises que permitam acompanhar a manifestação e mudança desses fenômenos ao longo do tempo. Este capítulo apresenta uma

visão geral de todo o processo de análise de textos de mídias sociais por meios computacionais, desde a extração de dados das mídias sociais e as principais técnicas de análise textual até exemplos de aplicações. Apesar deste capítulo ter foco em textos de mídias sociais e suas particularidades, boa parte das técnicas apresentadas podem ser também aplicadas em outras fontes textuais.

2.1. Introdução

Mídias sociais, tais como o Twitter, Reddit e Facebook são acessadas por aproximadamente 4,7 bilhões de pessoas em todo o planeta (i.e., 59% da população) [Kemp 2022]. Esses usuários, normalmente, engajam nas plataformas por muitas horas mensais e produzem e compartilham quantidades massivas de dados [Kemp 2022]. Por exemplo, anualmente os usuários do Twitter fazem cerca de 200 bilhões de postagens, o equivalente à 6 mil *tweets* por segundo [Stats 2022]. Aliado a isso, a maioria dessas mídias sociais disponibilizam meios para permitir o acesso programático a dados públicos em larga escala, seja por meio de APIs (*Application Program Interfaces*), aplicações de terceiros ou iniciativas de dados abertos (como descrito na Seção 2.2). Desta maneira, dados de mídias sociais têm sido amplamente utilizados por diversos estudos de fenômenos sociais, não somente no ambiente virtual onde são produzidos, mas também sendo traduzidos para o mundo real, uma vez que as mídias sociais implementam mecanismos que simulam interações sociais que acontecem *offline* [Silva et al. 2019]. Com isso, é possível inferir comportamentos *offline* a partir de "rastros" deixados de forma orgânica pelos usuários.

Nesse sentido, muito mais do que uma extensão do mundo real, tais mídias sociais podem ser consideradas como valiosas fontes de dados, sendo possível inferir comportamentos sobre sociedade urbanas em larga escala [Rogers 2009, Silva et al. 2019], contribuindo para diversas áreas de estudo, tais como a sociologia, psicologia, política, jornalismo, linguística, urbanismo, entre outros [Barbier and Liu 2011, Silva et al. 2019]. A indústria também tem tirado proveito de dados de mídias sociais para diversos propósitos, por exemplo, para conhecer melhor o perfil de clientes com maior potencial de compra, fazer comparação de desempenho com marcas concorrentes (*benchmarking*), inferir tendências de consumo (*forecasting*), personalizar recomendações de produtos e serviços, customizar e direcionar suas campanhas de comunicação de acordo com o perfil do cliente, entre várias outras possibilidades.

Dentre os possíveis tipos de mídia (texto, áudio, imagem, etc.) presentes em dados de mídias sociais, destaca-se a utilização de textos escritos em linguagem natural, por serem amplamente disponíveis na maioria delas. Por essa razão, neste capítulo apresentamos fundamentos teóricos de Processamento de Linguagem Natural (NLP, do acrônimo em Inglês de *Natural Language Processing*), aplicados aos desafios científicos e tecnológicos de lidar com textos gerados por usuários de mídias sociais, que podem ser explorados para desenvolver aplicações que se beneficiam do conhecimento extraído de tais textos. Como muitas técnicas, métodos e modelos de NLP são restritos de acordo com o idioma do texto, temos como foco neste capítulo textos de mídias sociais escritos em língua inglesa, por ser o idioma predominante na área de NLP e possuir o maior conjunto de recursos disponíveis. No entanto, sempre que possível, também são destacados recursos que são multilíngues, ou independentes de idioma, os quais podem ser explorados também em textos escritos em outros idiomas. Vale ressaltar também que, apesar do foco ser

em texto de mídias sociais, este material não é limitado somente a este tipo de conteúdo.

Com isso, este capítulo visa preparar o(a) leitor(a) para conhecer:

- As principais características de textos compartilhados por usuários em famosas mídias sociais, como coletá-los, assim como metainformações e respectivos desafios e limitações que cada uma dessas fontes de dados apresenta (Seção 2.2).
- Diversas técnicas para preparar os dados textuais, antes deles serem efetivamente utilizados por algum modelo de linguagem (Seção 2.3).
- Representações de textos utilizando vetores numéricos, chamados *embeddings*, capazes de capturar regularidades sintáticas e semânticas presentes nos textos (Seção 2.4).
- Diferentes métodos de modelagem e extração de conhecimento, incluindo técnicas de agrupamento de textos (clusterização), bem como a modelagem de estruturas semânticas latentes no texto, conhecida como extração de tópicos (Seção 2.5).
- Os conceitos de compreensão semântica e emocional, que abragem as tarefas de detecção de intenções, reconhecimento de entidades nomeadas, análises de sentimentos e emoções, ressaltando os desafios inerentes a elas (Seção 2.6).
- Aplicações reais, tais como a recomendação de rotas personalizadas para cidades inteligentes e o caso de uso na análise de situações politicamente polarizadas, que podem ser desenvolvidas com base no conhecimento semântico extraído a partir de dados de mídias sociais, considerando a utilização de diferentes técnicas apresentadas no minicurso (Seção 2.7).

Por fim, a Seção 2.8 traz as conclusões a respeito do tema.

2.2. Textos de mídias sociais: suas principais características e como coletá-los

Dados textuais em mídias sociais são conteúdos geralmente presentes em postagens e comentários, que podem tomar diferentes formas dependendo de como se dão as interações na respectiva plataforma em que foram produzidos, suas limitações e convenções. No Twitter[1], por exemplo, a plataforma permite um certo grau de anonimidade e as postagens são limitadas a 280 caracteres, por isso é comum que os usuários usem abreviações e adotem um tom informal, usando jargões e gírias comuns na *Internet*, incluindo muitas vezes construções ambíguas marcadas pelo sarcasmo [Tufekci 2014]. O Facebook[2], por sua vez, é uma mídia focada em interações entre pessoas reais, portanto, o não anonimato é uma característica marcante, ainda que haja incidência de contas falsas, muitas vezes focadas na manipulação da opinião pública [Weedon et al. 2017]. Já o Reddit[3], é uma mídia construída em torno de comunidades auto-organizáveis, onde os próprios membros ou moderadores são responsáveis por criar e executar as regras de moderação, portanto, a forma de interação depende diretamente da cultura de seus membros

[1]https://twitter.com. Último acesso em 05 de Agosto de 2022.

[2]https://facebook.com. Último acesso em 05 de Agosto de 2022.

[3]https://reddit.com. Último acesso em 05 de Agosto de 2022.

[Proferes et al. 2021]. Além de dados textuais, também é possível obter seus respectivos metadados, como informações pessoais do usuário, localização geoespacial de onde foi realizada a postagem, sinais sociais (como curtidas, votos ou reações), entre outros, que podem ser úteis em diferentes tipos de aplicações.

Nesta seção, são apresentadas as principais características de dados textuais disponíveis publicamente nas mídias sociais Twitter, Reddit e Facebook. Também são descritos os mecanismos de interação presentes em cada uma delas, os métodos para coletar dados em larga escala, o dicionário de dados e metadados disponíveis e, por fim, as limitações existentes.

2.2.1. Twitter

O Twitter é uma rede social *online* de *microblogging*, na qual os usuários podem postar mensagens com limite máximo de 280 caracteres, chamadas *tweets*. A plataforma do Twitter foi uma das primeiras a permitir acesso programático a dados públicos em larga escala por meio de API (acrônimo de *Application Program Interface*). Esse fato, tornou a utilização do Twitter bastante atrativo no meio acadêmico e na indústria [Tufekci 2014], dada a facilidade em se obter dados sobre eventos em tempo real ou, inclusive, dados históricos. Diferente de outras plataformas, como o Facebook, a maior parte dos dados do Twitter estão disponíveis abertamente na *web*, exceto informações de perfis privados (menos de 10% do total) ou mensagens privadas [Tufekci 2014].

Uma característica proeminente desta rede social é sua simplicidade, contando com algumas poucas funcionalidades, como (a) *hashtags*, palavras precedidas do símbolo #, usadas para demarcar um tópico de discussão, (b) *retweets*, ação de compartilhamento de mensagens, (c) menções, nome de usuários precedidos do símbolo @, quando um usuário marca outro em suas mensagens [Tufekci 2014], e (d) respostas, quando um usuário responde um *tweet*. Essas funcionalidades formam a base de como se dá a interação entre usuários dentro do Twitter e que possibilitam uma série de aplicações. Por exemplo, *hashtags* podem ser usadas para mapear assuntos importantes em um determinado local ou contexto, podendo representar a opinião de parte da população e, assim, sendo útil para planejamento de políticas públicas [Cody et al. 2015]. Outro exemplo, é a análise da rede de *retweets*, onde pode ser compreendido o comportamento de indivíduos em grupos homófilos e em uma situação politicamente polarizada, quando estes são expostos a conteúdos contrários ao seu viés político [Kobellarz et al. 2022]. A rede de menções, por sua vez, também pode ser uma importante fonte de informação em contextos políticos. Por exemplo, em [Conover et al. 2011], os autores identificaram que a rede de *retweets* em uma situação polarizada pode apresentar um alto grau de segregação entre grupos políticos opostos, enquanto a rede de menções não apresenta um padrão claro que possa ser ligado ao viés político.

Além dos dados textuais contidos nos *tweets*, também é possível obter seus metadados[4], tais como: o *timestamp* da postagem; os dados do perfil do usuário que fez a postagem, incluindo seu nome, localização, se é uma conta verificada pelo Twitter, quantidade de seguidores, amigos e postagens, língua do perfil, entre outros; as coordenadas

[4]https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model/tweet. Último acesso em 05 de Agosto de 2022.

geográficas do *tweet* no formato GeoJson[5], que são reportadas pelo próprio usuário ou obtidas automaticamente pelo dispositivo que gerou o *tweet*; a associação do *tweet* a um determinado local (cidade, estado ou país); a quantidade de vezes que o *tweet* foi citado, respondido, compartilhado (*retweet*) ou curtido; e, as entidades presentes no *tweet*, como *hashtags*, *links*, usuários mencionados, endereço para mídia anexada na postagem, entre outros. Caso o *tweet* seja uma resposta a outro *tweet* ou um *retweet*, o identificador único da mensagem original e do seu respectivo autor são incluídos entre os metadados, permitindo obter facilmente os dados do *tweet* original programaticamente.

Data a vasta quantidade de pesquisas envolvendo dados do Twitter, suas limitações são bem conhecidas na literatura [Tufekci 2014]. A primeira é sobre o limite máximo de 280 caracteres por *tweet*, que pode restringir a capacidade argumentativa dos usuários em discussões. Essa limitação muitas vezes é contornada com a utilização de contrações de palavras, gírias da *Internet* e *emojis*, tornando complexa a tarefa de análise desses textos [Tufekci 2014]. Outra limitação é a representatividade dos dados, uma vez que a API do Twitter retorna dados parciais considerando sua relevância para uma dada consulta (como descrito na documentação[6]). Por isso, não é possível obter todos os *tweets* para uma consulta específica, comprometendo a replicabilidade do processo de obtenção de dados. Nesse sentido, a escolha do critério de filtragem deve considerar que a amostra não seja resultado da auto-seleção, ou seja, os critérios de filtragem não devem ser escolhidos com a intenção de forçar um resultado esperado por quem fez a seleção [Tufekci 2014]. Isto é uma implicação ética importante em estudos que usam dados do Twitter. Outra característica notável dessa fonte de dados é a incidência de contas robô, que em alguns casos podem comprometer a conclusão de resultados. Por exemplo, é reconhecido que a simulação de comportamentos simples no Twitter, como seguir contas e retuitar conteúdos de outros usuários, com o intuito de ser seguido de volta ou ser retuitado, são moedas sociais que podem tornar um robô tão influente quanto contas reais [Messias et al. 2013].

2.2.2. Reddit

O Reddit é uma mídia social composta por comunidades de discussão sobre temas específicos no formato de fóruns, chamados "*subreddits*". Em 2020, a plataforma tinha mais de 100 mil comunidades com 50 milhões de usuários ativos diariamente[7]. O uso de dados dessa plataforma se tornou bastante atrativo no meio acadêmico, uma vez que adotou um modelo similar ao do Twitter, disponibilizando dados abertamente por meio de uma API. Uma das características proeminentes são os mecanismos de moderação dos *subreddits*, que geralmente possuem regras claras que devem ser seguidas pelos seus membros, sob risco de serem penalizados pelos responsáveis pela comunidade, assim como por outros membros que podem votar nos comentários mais relevantes. Além da moderação, mecanismos de recompensa possibilitam a bonificação de membros que respeitam as regras e contribuem ativamente nas comunidades. Essas características, em conjunto, permitem uma maior riqueza nos dados tanto quantitativamente quanto qualitativamente [Proferes et al. 2021].

[5]https://geojson.org. Último acesso em 05 de Agosto de 2022.

[6]https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/overview. Último acesso em 05 de Agosto de 2022.

[7]https://www.redditinc.com/advertising/audience. Último acesso em 05 de Agosto de 2022.

A API do Reddit permite acesso à qualquer informação disponível publicamente na plataforma, incluindo postagens, comentários, perfis, comunidades e suas respectivas metainformações. É possível capturar dados históricos e em tempo real, similar a API do Twitter, mas com a vantagem de permitir a recuperação do histórico completo, diferentemente do Twitter que foca na relevância - essa característica torna o procedimento de coleta de dados no Reddit replicável, algo desejável em pesquisas acadêmicas. O dicionário de dados do Reddit também é bastante rico, por exemplo, além do conteúdo da postagem, é possível recuperar todos os comentários incluindo seu conteúdo completo e respectivos metadados, com informações do autor (nome, pontuação na plataforma, se possui e-mail verificado, entre outros[8]), informações sobre a pontuação do comentário, assim como a quantidade de votos positivos e negativos; *flag* apontando se o comentário foi editado pelo usuário ou destacado pelos moderadores; a árvore de respostas em cascata gerada por um comentário; entre outros dados.

Dentre os desafios ao trabalhar com dados do Reddit, está o fato da plataforma permitir um alto grau de anonimidade, possibilitando que usuários mal-intencionados se comportem de forma antiética em comunidades que tenham regras menos restritivas [Proferes et al. 2021]. Por exemplo, para fazer parte da plataforma, basta cadastrar um nome de usuário e e-mail, sem a necessidade de verificação, inclusive, é encorajado o uso de pseudônimo para manter a privacidade do perfil [Proferes et al. 2021]. Outro desafio está relacionado às comunidades serem regidas pelos seus próprios membros, consequentemente, culminando em práticas de moderação distintas. Isso torna complexa a comparação entre comunidades, mesmo utilizando os mesmos critérios para estudar um fenômeno em comum [Proferes et al. 2021]. Quanto aos conteúdos em texto, a liberdade dada no formato, que pode inclusive ter *tags Markdown* ou *html*, ao mesmo tempo que trazem riqueza de detalhes, também tornam a limpeza dos dados mais complexa. Outra característica limitante é o fato de *bots* terem uma grande representatividade em algumas comunidades, podendo atuar como criadores e curadores de conteúdos e, inclusive, como moderadores para automatizar a aprovação de postagens em *subbreddits*.

2.2.3. Facebook (Meta)

O Facebook, um dos produtos da Meta, é uma rede social pautada em interações entre pessoas reais. Diferentemente do Reddit e Twitter, o alto grau de controle de privacidade das contas nessa rede é uma característica marcante e o anonimato é desencorajado. Medidas restritivas são aplicadas para coibir comportamentos nocivos na plataforma, como os ligados à manipulação da opinião pública por meio do compartilhamento de informações falsas ou desinformação [Weedon et al. 2017]. Apesar da plataforma do Facebook disponibilizar APIs públicas para a criação de aplicações, o acesso aos dados é limitado àqueles sobre os quais o usuário autenticado tem permissões para gerenciar, não sendo possível a extração de dados públicos em larga escala.

Alternativamente, é possível coletar dados sem usar uma das APIs oficiais do Facebook, por exemplo, por meio do programa *Social Science One*[9], que permite estabelecer uma parceria entre instituições de ensino e o Facebook, para ter acesso direto à sua base

[8]https://praw.readthedocs.io/en/stable/code_overview/models/comment.html. Último acesso em 05 de Agosto de 2022.

[9]https://socialscience.one/grant-process. Último acesso em 05 de Agosto de 2022.

de dados. Apesar disso, a aplicação no programa demanda uma série de etapas legais, além de impor limitações significativas no nível de dados que podem ser acessados. Outra possibilidade, é realizar o recrutamento de voluntários para preencherem pesquisas ou cederem seus dados, utilizando aplicações de terceiros que se conectam às APIs oficiais. Esse método apresenta uma restrição considerável na abrangência do perfil e quantidade de usuários, limitando os dados aos usuários que foram recrutados e cederam seus dados para pesquisa. A raspagem de dados (ou, *scraping* em Inglês) também é uma das alternativas para coleta de dados, contudo esse procedimento é explicitamente proibido pelas políticas da plataforma, assim como é desencorajado por meio de sistemas *anti-bot* que limitam esse método.

Recentemente, uma iniciativa da Meta chamada CrowdTangle[10], permitiu a pesquisadores e empresas terem acesso a grande parte dos dados abertos do Facebook de forma gratuita, incluindo postagens de todas as páginas públicas com pelo menos 50 mil curtidas, grupos com pelo menos 95 mil membros, grupos dos Estados Unidos com pelo menos 2 mil membros e de todos os perfis verificados na plataforma. Por meio do CrowdTangle, também é possível coletar dados do Instagram, outra rede social da Meta, focada em postagens no formato de imagem e vídeo. Por brevidade, focaremos a seguir somente na rede social do Facebook. Dentre os dados que podem ser coletados do Facebook por meio do CrowdTangle estão: a postagem em texto, imagem ou vídeo, *timestamp* da postagem, tipo da postagem (vídeo, imagem ou texto), a página, perfil ou grupo em que foi feita a postagem, contagem de interações sociais (i.e., a quantidade de curtidas, reações segmentadas por tipo, comentários, compartilhamentos e visualizações), e quais páginas públicas ou perfis compartilharam a postagem. Mais informações podem ser encontradas na documentação da ferramenta[11].

Os dados do CrowdTangle podem ser obtidos programaticamente via API ou por meio da interface gráfica da própria ferramenta, que também permite a criação de *dashboards* personalizados para análise de postagens em tempo real. Seu sistema de busca é bastante robusto, permitindo desde uma busca básica por *hashtags* ou ocorrência de palavras-chave em textos de postagens, até busca textual em imagens. Apesar disso, o nível de informação retornado é limitado: diferentemente do Twitter e Reddit, não é possível obter informações sobre quem reagiu ou visualizou uma postagem, assim como nenhuma informação demográfica sobre usuários, como idade ou localização. Comentários em postagens também não estão disponíveis. Outra limitação, é o fato dos dados que podem ser obtidos serem produzidos apenas por contas famosas ou verificadas, o que inviabiliza o estudo do comportamento de postagem de públicos menos representativos no Facebook. Todas essas limitações são esforços focados na proteção da identidade dos usuários, característica fundamental dessa fonte de dados, aliás, casos emblemáticos de violação da privacidade de usuários do Facebook ficaram bastante conhecidos, como o do *"Tastes, ties, and time"* ou 3T e o da *Cambridge Analytica*, duas situações nas quais, mesmo anonimizando a identidade dos usuários, foi possível reconstituir as informações e expor seus perfis [Zimmer 2020].

[10]https://crowdtangle.com. Último acesso em 05 de Agosto de 2022.

[11]https://help.crowdtangle.com/en/articles/4201940-about-us. Último acesso em 05 de Agosto de 2022.

2.3. Pré-processamento Textual

Modelos computacionais voltados à compreensão de linguagem natural não são capazes de reconhecer o significado abstrato que atribuímos às palavras que conhecemos, tampouco seus atributos semânticos, culturais ou históricos. Normalmente, tais modelos são projetados para inferir o significado das palavras e expressões por meio da *distribuição estatística* em que ocorrem em um texto, ou seja, as janelas contextuais e a frequência em que ocorrem.

Para que isso ocorra, cada item léxico é tratado como um *índice* contendo um valor, dentro de um *vetor* numérico. Este vetor, por sua vez, representa o contexto em que tal palavra está inserida (uma sentença ou documento, por exemplo), e o valor numérico atribuído a cada palavra pode indicar, dentre outros: a ocorrência ou não de determinadas palavras de interesse em um documento, a sua distribuição relativa em um conjunto de vários documentos ou, até mesmo, a relação léxica entre palavras e sentenças.

A etapa de pré-processamento é necessária para garantir que, ao ser atribuída uma representação numérica para cada palavra de uma sentença, mantenhamos apenas os conteúdos e características desejáveis para a análise computacional do *corpus* com o qual se trabalhará. Desta forma, a complexidade do domínio do problema é reduzida, eventuais ruídos podem ser removidos, aumentando as chances de sucesso do projeto como um todo. Não existe, no entanto, uma fórmula única e precisa que funcione em qualquer projeto ou contexto: as seções a seguir apresentam, como referência, as técnicas mais comuns e amplamente utilizadas pela academia e indústria para preparação dos dados textuais, antes de serem efetivamente analisados por algum modelo computacional. Deve-se considerar que o emprego ou não de cada técnica deve levar em conta uma análise profunda do conteúdo textual a ser investigado.

O pré-processamento, normalmente, é a segunda etapa em um projeto de NLP, vindo logo após a obtenção dos dados. Nessa etapa os dados textuais são estruturados considerando o domínio e método que será aplicado.

2.3.1. Compreensão do domínio

Antes de qualquer tarefa de manipulação de dados textuais, é importante que o domínio de origem do texto seja bem compreendido, e que se tenha clareza dos objetivos pretendidos com a sua análise através de um modelo computacional. Em relação ao domínio de origem: textos de notícias, por exemplo, tendem a adotar uma linguagem formal, com poucos marcadores de expressão (i.e., exclamações, interrogações, repetições, etc.), enquanto textos de redes sociais podem conter gírias particulares, *emojis* e *hashtags* que podem ser relevantes para o objetivo do projeto. Caso um modelo computacional seja aplicado sobre textos do primeiro caso, caracteres especiais poderiam ser removidos sem impacto na compreensão de seu conteúdo. No segundo caso, no entanto, a simples remoção de caracteres especiais poderia ser bastante prejudicial, pois acabaria removendo outros elementos que contribuem na compreensão desse tipo de texto.

Além da compreensão do domínio, o entendimento do método que será aplicado é de vital importância na escolha das técnicas de pré-processamento. Por exemplo, em um projeto no qual o objetivo é identificar postagens similares ou tópicos latentes em um conjunto de postagens, é importante que a etapa de pré-processamento preserve palavras

representativas para a identificação de tópicos e elimine aquelas que possam trazer ruído. Nesse caso, é importante destacar palavras que ajudem a inferir regularidades entre as postagens, e remover ou penalizar outras que ocorram com alta frequência no conjunto de dados, mas que não colaboram na tarefa.

Outro exemplo seria a análise de sentimentos ou identificação de toxicidade em postagens. Em casos mais simples, o sentimento ou a toxicidade pode ser inferida por meio do uso de dicionários que mapeiam palavras às suas respectivas valências (i.e., polaridade negativa ou positiva). Nesse tipo de aplicação, pontuações, palavras comuns na gramática da língua do texto e qualquer outra palavra que não possua ocorrência no dicionário, poderiam ser removidas sem impactar no resultado. Contudo, esse método não captura sutilezas que podem ser importantes em determinados tipos de aplicações, como, por exemplo, quando é necessário que o modelo contemple a relação entre palavras ou frases para identificar sentimentos ou características tóxicas em conteúdos compostos por sarcasmo ou figuras de linguagem. Nesse caso, boa parte das características textuais precisam ser preservadas para a aplicação de modelos de aprendizado de máquina modernos, que consigam capturar essas relações no texto (alguns modelos serão apresentados no decorrer das seções 2.4 e 2.5).

2.3.2. *Tokenização*

Até essa seção, nos referimos às unidades básicas presentes em textos usando o termo "palavras", "símbolos", entre outros. Apesar disso, existe uma designação apropriada no contexto de NLP para se referir de forma genérica a essas unidades: *tokens*.

O termo *token*, derivado das linguagens formais de programação, consiste em uma sequência de caracteres dentro de um documento textual que apresente algum tipo de unidade semântica [Schütze et al. 2008]. No contexto da NLP, um *token* pode ser representado por um caractere ou uma sequência de caracteres, ou mesmo um símbolo, pontuação, número, *emoji*, *hashtag*, menção, entre outros. O processo usado para transformar um documento textual em uma lista composta por esses elementos é chamado de *tokenização*.

Considere um *corpus* formado pelas últimas palavras de Sócrates, segundo relato de Platão no diálogo *Fédon* (retirado de [Pombo 2022]):

> *"Críton, somos devedores de um galo a Asclépio; pois bem pagai a minha dívida, pensai nisso"*

Após o processo de *tokenização* da sentença original obteríamos a seguinte lista de *tokens*:

> *["Críton", ",", "somos", "devedores", "de", "um", "galo", "a", "Asclépio", ";", "pois", "bem", "pagai", "a", "minha", "dívida", ",", "pensai", "nisso"]*

2.3.3. Normalização de texto

Após a transformação do *corpus* em uma lista de *tokens*, passamos por uma série de tarefas cujo objetivo é reduzir ao mínimo possível a complexidade do vocabulário com o qual trabalharemos. Isto é importante para aumentar a probabilidade de que o método aplicado nas etapas posteriores tenha maior chance de êxito na tarefa proposta.

Reduzir a variabilidade do *corpus* envolve, por um lado, remover termos que não possuem nenhum tipo de relevância para o domínio analisado (o que discutiremos nas etapas a seguir), e por outro, agrupar elementos idênticos ou similares, que estejam apresentados de forma distinta. Uma das formas mais simples e efetivas de fazê-lo é através da normalização da caixa das palavras presentes no vocabulário. Este processo consiste em transformar todas as palavras em minúsculas ou maiúsculas. Trata-se de uma funcionalidade computacionalmente muito simples, normalmente incorporada como uma função nativa das principais linguagens de programação (como o *.lower()* e *.upper()*, do *Python*, e o *.toLowerCase()* e *.toUpperCase()*, do *Javascript*, que uniformizam os *tokens* do *corpus* em palavras minúsculas ou maiúsculas, respectivamente).

Suponha que, analisando textos de redes sociais num domingo à tarde, nos deparemos com comentários a respeito de resultados de uma partida de futebol: um torcedor mais empolgado pode comemorar a pontuação de seu time escrevendo "GOL !!", enquanto outro, desapontado ao final da partida, escreve apenas "Perdemos por causa de um gol". Ou que ainda um outro, atendo-se apenas aos fatos, escreva "Gol aos 45 do segundo tempo". A mesma palavra, então, assume três grafias distintas ("GOL", "Gol", "gol"). Caso estes exemplos sejam submetidos a um algoritmo computacional da maneira como originalmente se apresentam, serão consideradas por este algoritmo como três *tokens* distintos. O processo de normalização fará com que este sistema computacional possa reconhecer, devidamente, as três como ocorrências de um único *token*, o que lhe permitirá atribuir a relevância devida deste *token* na compreensão do conteúdo textual analisado.

Há de se considerar, no entanto, que dada a natureza expressiva dos discursos presentes nas redes sociais, é possível que, além das exemplificadas, outras grafias possam ser adotadas ("GOOOOL!", por exemplo). Estas, no caso, não são resolvidas simplesmente pelo processo de normalização. Trata-se de outro desvio que pode ser contornado em outros níveis de análise. Uma destas formas, discutida a seguir, seria através da aplicação de expressões regulares.

Retornando ao nosso exemplo, após o processo de normalização para letras minúsculas, obteríamos o seguinte resultado:

> *["críton", ",", "somos", "devedores", "de", "um", "galo", "a", "asclépio", ";", "pois", "bem", "pagai", "a", "minha", "dívida", ",", "pensai", "nisso"]*

2.3.4. Expressões Regulares (Regex)

Para prosseguir na limpeza e pré-processamento de texto, garantindo que nenhuma informação relevante seja perdida, cabe agora uma análise do domínio semântico coberto pelo *corpus* utilizado. Algumas perguntas que podem ajudar nesta análise: Qual a natureza do *corpus* a ser investigado? É composto por textos formais (e.g., documentos, artigos, etc.), ou informais (e.g., textos de redes sociais, fóruns, *chats*)?

Caso sejam textos em linguagem formal, de notícias por exemplo, não é comum que se encontre *emojis* ou erros de digitação, e toda pontuação é utilizada de maneira ponderada. Uma ferramenta que permita remover todos os caracteres especiais e mantenha apenas o texto pode ser muito útil. Caso sejam textos de postagens de mídias sociais,

pode ser necessário manter alguns padrões de caracteres que correspondam aos *emojis*, removendo apenas os *links* por exemplo. Toda manipulação (e.g., contagem, substituição ou remoção de padrões) de texto apenas se faz possível a partir de uma funcionalidade que permita, de maneira preliminar, *encontrar* tais padrões no texto: ao conjunto de comandos que permitem este nível de manipulação de texto se dá o nome de Expressões Regulares.

As Expressões Regulares (do inglês, *Regular Expressions*, cujos acrônimos são *RE* ou *RegEx*), são comandos que especificam padrões de busca em texto [Jurafsky and Martin 2021]. As expressões regulares foram propostas inicialmente na década de 1950 por [Kleene et al. 1956] como uma notação algébrica para a representação de eventos nos primeiros modelos de redes neurais computacionais de McCulloch-Pitts, tendo sido adaptadas para uma ferramenta de busca de texto em linguagem natural por [Thompson 1968]. É ainda hoje incorporado aos editores de texto dos principais sistemas operacionais. A maior parte das linguagens de programação, do C++ ao Python, possuem suporte para manipulação de texto a partir de expressões regulares.

Voltando ao exemplo relacionado à palavra *gol*: suponha que queiramos encontrar todas as ocorrências desta palavra dentro de um *corpus* de textos de redes sociais (já normalizado). Para isso, basta indicar este padrão dentro de duas barras invertidas: $/gol/$. Para encontrarmos todas as variações em que a letra *o* dentro da palavra venha ser repetida (como no exemplo "goool !"), acrescentamos ao nosso padrão um quantificador $*$ à letra *o*: $/g[o]*l/$. Agora, suponha que estejamos lidando com um *corpus* de textos do *Twitter*, e queiramos encontrar todas as referências a *usernames* (iniciados por @), podemos utilizar o comando $/@(\backslash w\{1,15\})/$, que traduzindo, significaria: buscar elementos iniciados por @, seguidos por qualquer sequência entre 1 a 15 caracteres alfanuméricos. Uma vez que identificados, os *usernames* particulares podem ser substituídos por um *token* único.

As bibliotecas de manipulação de Expressões Regulares, como a *RE* do Python, permitem facilmente a manipulação de texto. No caso do *Twitter*, sabemos que as menções a usuários são sempre precedidas de "@", e que as *hashtags* são iniciadas por "#". Neste caso, a substituição dos *tokens* correspondentes a nomes de usuários por um *token* único (como, por exemplo, *@USUARIO*), permite melhorar a qualidade da análise computacional de texto, por permitir contar a quantidade de referências a usuários, independentemente de quem seja, analisar os contextos de sentenças em que referências a usuários aparecem e, ao mesmo tempo, anonimizando a identidade real destes, o que garante a privacidade dos dados analisados.

Existem algumas classes de comandos de Expressões Regulares, como os marcadores, os quantificadores e as âncoras. Os marcadores são aqueles responsáveis por encontrar determinados padrões num documento textual; os quantificadores, por sua vez, permitem que se especifique a quantidade de vezes que tais padrões devem ser buscados. Já as âncoras, são os delimitadores de início e fim de onde tais padrões devem ser encontrados.

Abaixo, alguns exemplos de comandos marcadores (adaptado de [Jurafsky and Martin 2021]). Em sequência, alguns exemplos de quantificadores e âncoras, que combinados aos marcadores, permitem que se especifique a quantidade e as condições em que os padrões devem ser encontrados:

Padrão	Descrição
\w	Qualquer símbolo alfanumérico
\W	Qualquer elemento não-alfanumérico
\s	Espaçamento (Espaço em branco)
\S	Espaço que não esteja em branco
\d	Qualquer dígito
\D	Qualquer símbolo diferente de dígito

Padrão	Descrição
()	Parênteses delimitando uma sub-expressão que deve ser encontrada
* + ? {}	Contadores (número de vezes em que a expressão anterior deve se repetir)
^$	Indicação de início e fim de sentença
|	Disjunção ("Ou"), indicando que um dos conjuntos deve ser encontrado

Expressões regulares, no entanto, podem se tornar muito complexas pelo mesmo motivo que são versáteis: o fato de poderem ser combinadas de maneira irrestrita e ilimitada. Não é necessário, contudo, decorar padrões de expressões regulares: diversas páginas, como o Regex101 [Dib 2022] possuem guias de referência e ambiente para montar e testar os padrões que possam ser úteis em cada caso

Retornando ao nosso exemplo da clássica enunciação em *Fédon*, suponhamos que queiramos manter apenas as palavras, removendo toda a pontuação e caracteres especiais. Utilizando o comando $[\backslash w+]$, obteríamos o seguinte conjunto de *tokens* como resultado:

> *[“críton”, “somos”, “devedores”, “de”, “um”, “galo”, “a”, “asclépio”, “pois”, “bem”, “pagai”, “a”, “minha”, “dívida”, “pensai”, “nisso”]*

2.3.5. Remoção de *stop-words*

Por causa da natureza da sintaxe da língua natural humana, quando nos comunicamos, fazemos uso de dois tipos de palavras: as palavras que carregam o conteúdo de determinada mensagem que queremos transmitir (os verbos, substantivos, adjetivos e advérbios); e as palavras de função, que não carregam por si significado relevante, mas são necessárias para a manutenção da gramaticalidade daquilo que se enuncia. Nesta segunda categoria, se encontram, por exemplo, os artigos, as conjunções e as preposições.

Uma análise da frequência (o número de ocorrência de palavras individuais em um dado texto), normalmente, aponta para o fato de que as palavras de função ocorrem com muita frequência, pelo fato de estarem presentes em quase todos os enunciados, independente de seu conteúdo. Trata-se de um padrão que pode ser observado em praticamente todos os idiomas da língua humana (inclusive, trata-se de um dado provado empiricamente, por aquela que ficou conhecida como Lei de Zipf - ler mais em [Zipf 2016]).

As palavras de conteúdo, por sua vez, são mais raras, e muito associadas à mensagem que se deseja transmitir. Num algoritmo para análise de sentimentos, por exemplo, pode ser interessante encontrar palavras que carreguem uma conotação positiva, que embora raras, podem estar associadas a elogios - enquanto palavras de conotação negativa podem estar associadas a críticas.

Remover ou não *stop-words* é uma decisão que deve levar em conta não mais o *corpus*, mas principalmente o algoritmo a ser utilizado para classificação ou análise. A remoção de *stop-words* é muito associada a algoritmos de aprendizado de máquina supervisionados mais tradicionais, como Naïve Bayes e Regressão Logística, que podem levar em conta a frequência dos termos presentes no *corpus* para determinar a classe a qual cada texto pertence, ou métodos de representação baseados em *Bag Of Words* (sobre os quais falaremos na próxima Seção).

Um exemplo de caso em que a remoção de *stop-words* não se faz necessária: para algoritmos baseados em redes neurais recorrentes, como as do tipo LSTM (*Long-Short Term Memory*), por exemplo, a remoção de *stop-words* pode não vir a se fazer necessária, pois o aprendizado é baseado na análise da sequência de palavras - e a termos muito frequentes são automaticamente atribuídos pesos baixos para a computação de significado.

Caso se decida pela remoção de *stop-words*, é necessário um cuidado em especial: a palavra "não" está geralmente contida em listas de *stop-words* a serem removidas (estando, por exemplo, nas *stop-words* nativas do português na biblioteca NLTK, do Python [Bird 2006]). No entanto, a remoção de um "não" em uma sentença pode, literalmente, inverter o seu significado. Portanto, uma boa prática é manter as negações ainda que se removam as outras *stop-words*.

Retornando ao nosso exemplo, caso queiramos remover as *stop-words* da sentença, manteríamos a seguinte lista de *tokens*:

> *["críton", "devedores", "galo", "asclépio", "pois", "bem", "pagai", "dívida", "pensai", "nisso"]*

Nesse caso, os *tokens* removidos foram:

> *["somos", "de", "um", "a", "a", "minha"]*

2.3.6. Lematização e Stemização

A lematização e a *stemização* (adaptados dos termos em inglês, *lemmatization* e *stemming*) são outros dois processos para diminuição da complexidade e variabilidade de um *corpus*. Ambos consistem em análises léxicas feitas com o auxílio de dicionários específicos da língua no *corpus* ao qual serão aplicadas.

A lematização consiste num processo para encontrar o *lema* de cada palavra de um *corpus*, ou seja, sua forma morfológica semântica e sintaticamente canônica [Indurkhya and Damerau 2010]. A lematização segue a premissa de que, ao se reduzir todas as inflexões possíveis que as palavras podem assumir (como, por exemplo, em relação a tempos verbais, gênero, número ou grau), o vocabulário do *corpus* é simplificado, facilitando a capacidade preditiva dos modelos de aprendizado de máquina a serem aplicados. Através da lematização, todas as ocorrências de variações do verbo *estar* (como, por exemplo, "estou", "estaremos", "estará"), seriam substituídas pela forma canônica do verbo.

Já o processo de *stemização*, trata-se de um algoritmo que reduz as palavras ao seu *radical* (ou *tema*). Em relação ao nome do processo, opta-se por manter a denominação derivada do inglês apenas para facilitar a referência a literatura relacionada em domínio computacional. No processo de *stemização*, são eliminados quaisquer afixos

agregados ao radical de uma palavra dentro de uma sentença (prefixos ou sufixos). A *stemização* é feita a partir de algoritmos de busca, que procuram pela ocorrência de determinados radicais que podem ocorrer em cada palavra. Trata-se, portanto, de um algoritmo mais simples, que apresenta algumas desvantagens em relação à lematização [Indurkhya and Damerau 2010], por reduzir a legibilidade, e, ao mesmo tempo, ser mais "agressivo" e, por vezes, excessivamente generalizador. Por exemplo, o *Porter Stemmer* [Porter 1980] reduz *organization* para *organ*) [Jurafsky and Martin 2021]. Esta imprecisão se deve, em parte, por desconsiderar que, devido ao caráter dinâmico e evolutivo das línguas, existam palavras que compartilhem um radical, mas ao longo do tempo ganhem significados próprios incompatíveis entre si.

Algoritmos de lematização e *stemização* partem de um processo heurístico, específico a cada linguagem. Ambas as tarefas possuem implementações no português brasileiro (como o lematizador LemPORT [Rodrigues 1997] e o RSLP Stemmer do NLTK [Bird 2006]), mas pode ser uma limitação quando aplicado a idiomas que possuam poucos recursos computacionais disponíveis.

Retornando ao exemplo, ao aplicarmos um lematizador, obteríamos o seguinte resultado sobre os *tokens* em questão:

> *[("críton", "críton"), ("devedores", "devedor"), ("galo", "galo"), ("asclépio", "asclépio"), ("pois", "pois"), ("bem", "bem"), ("pagai", "pagar"), ("dívida", "dívida"), ("pensai", "pensar"), ("nisso", "nisso")]*

2.3.7. N-gramas

Os *n-gramas* são, basicamente, sequências de *n* palavras. Por exemplo, caso queiramos capturar todas as sequências de duas palavras (2-gramas, ou "bigramas") em um enunciado como "eu quero água gelada", obteríamos as expressões:

> *<"eu, quero">, <"quero, água">, <"água, gelada">*

A análise de *n-gramas* pode ser bastante útil para entender as formas de expressão mais frequentes dentro de um *corpus*. Ferramentas de *n-gramas*, estão por trás das técnicas mais tradicionais para produção de modelos de linguagem, que permitem estimar a probabilidade de ocorrência de uma palavra em uma sequência, que é muito difundida em sistemas de reconhecimento de fala e para sugestão de digitação nos teclados de celulares.

Trata-se, portanto, de uma técnica bastante simples, que, ao mesmo tempo, facilita uma análise do comportamento agregado do conteúdo de um volume grande de texto. Em alguns casos, pode ser utilizado como uma tarefa adicional de limpeza de dados, onde a análise da frequência dos *n-gramas* de um determinado texto, pode ser utilizado como mais uma ferramenta de limpeza possível e, assim, sequências muito frequentes podem ser tratadas como *stop-words*, e sequências muito raras podem ser tratadas como ruído. O uso, mais uma vez, deve partir de uma análise do contexto do *corpus* no qual empregado.

2.3.8. *POS-Tagging*

As *POS-Tags* (do inglês *Part-of-Speech*, ou Partes da Fala em tradução livre), realizam a categorização da função gramatical de cada palavra na sintaxe de uma sentença. Em

outras palavras, qual o "papel" de cada palavra na construção de uma sentença. São as categorias como adjetivos, verbos e substantivos.

A lista de *POS-Tags* mais comuns são (Adaptado de [Jurafsky and Martin 2021]): (*i*) ADJ: Adjetivo; (*ii*) ADP: Preposição (do inglês *Adposition*); (*iii*) ADV: Advérbio; (*iv*) NOUN: Substantivo; (*v*) VERB: Verbo; e, (*vi*) PROPN: Nomes próprios.

Existem duas classes principais de *POS-Tags*: as classes fechadas, que contém um número fixo e geralmente fechado dentro de uma língua, como as preposições e artigos; e as classes abertas, como os nomes e verbos, os quais são geralmente os mais afetados pela criatividade da língua humana, sendo impossível mapear todas as possibilidades de ocorrências de termos que podem caber nestas classes.

A obtenção das *POS-Tags* pode ser um indicador útil dependendo do contexto onde aplicado. Ainda que sejam atributos puramente gramaticais, as *POS-Tags* estão geralmente relacionadas à função semântica dos termos em uma frase. Por exemplo, em um contexto em que se deseje conhecer a impressão dos usuários em relação a um determinado produto ou serviço, pode ser útil manter aquelas palavras cujas *POS-Tags* indiquem serem adjetivos (rotulados como ADJ), ao passo que os substantivos (indicados como NOUN) podem ser úteis para uma etapa posterior de compreensão e limpeza do texto (através do reconhecimento de entidades nomeadas, assunto que será apresentado na Seção 2.6) [Jurafsky and Martin 2021, Indurkhya and Damerau 2010].

Existem diversos algoritmos para geração de *POS-Tags*, como *Hidden Markov Models* e *Conditional Random Fields*, sobre os quais se pode ler mais em [Lafferty et al. 2001]. Assim como em relação aos lematizadores, citados anteriormente, já existem implementações de algoritmos para detecção de *POS-Taggings* no português brasileiro [Honnibal et al. 2020, Bird 2006].

Essas seriam as classes gramaticais retornadas para cada *token* em nosso exemplo ao ser aplicado um algoritmo de *POS-Tagging*:

> *[("críton", "PROPN"), (",", "PUNCT"), ("somos", "AUX"), ("devedores", "ADJ"), ("de", "ADP"), ("um", "DET"), ("galo", "NOUN"), ("a", "ADP"), ("asclépio", "PROPN"), (";", "PUNCT"), ("pois", "ADV"), ("bem", "ADV"), ("pagai", "ADJ"), ("a", "DET"), ("minha", "DET"), ("dívida", "NOUN"), (",", "PUNCT"), ("pensai", "VERB"), ("nisso", "PRON")]*

2.3.9. Codificação de Caracteres

A forma escrita das línguas de origem latina (como o português e o espanhol) e anglo-saxônica (como o inglês), é baseada no alfabeto de 26 letras, tal qual utilizado neste texto. Ainda assim, existem variações da escrita do português e espanhol que não estão presentes no inglês, como a acentuação e a cedilha. Uma vogal acentuada, como "ã", é por sua vez, reconhecida por um sistema computacional como um caractere diferente de "a", e assim ocorrendo para qualquer outro acento ou variação que sofra.

Indo um pouco além, como se sabe, existem inúmeras outras alternativas de escrita além do alfabeto tal qual nos acostumamos. Podemos citar o alfabeto cirílico, o grego, as vogais vazias do norueguês, e ainda as formas de escrita do árabe e hebraico. Indo além,

o mandarim e o japonês, algumas das línguas mais faladas do planeta, não são baseadas em um alfabeto, mas em ideogramas, cuja variabilidade e significação é muito ampla do que as letras do alfabeto quando analisadas por si só.

Computacionalmente, cada um dos caracteres que pode compor uma língua deve ser representado em uma codificação única, compatível com a tipografia adotada. Considerando que a grande parte das ferramentas de processamento de linguagem natural é ainda na atualidade baseada no inglês, é muito comum que, ao inserirmos caracteres comuns ao português brasileiro, ocorra um erro de processamento ou perda do caractere, caso a codificação adotada seja incompatível com a aparição de caracteres acentuados.

Existe uma série de codificações possíveis, como Unicode, ASCII e UTF-8, que garantem a conversão correta entre o termo utilizado e um endereço de memória no sistema computacional. Portanto, é necessário utilizar uma codificação compatível com a língua sendo processada, para garantir o processamento correto pelos algoritmos que serão utilizados em seguida e evitar a perda de informação. Para conhecer mais a respeito de codificação, recomendamos a leitura de [Indurkhya and Damerau 2010].

Os métodos e técnicas descritos acima são, em geral, os mais utilizados e difundidos para o pré-processamento de texto. No entanto, diversos outros, mais específicos, podem ser aplicados a depender do contexto. Por exemplo, métricas de distância de edição, como o algoritmo de Levenshtein, para encontrar palavras parecidas e, através disso, detectar possíveis erros de digitação - ler mais em [González-Bailón and De Domenico 2021]; alguma versão do *Flesch Reading Score* do inglês, para remoção de eventuais palavras sem sentido, comuns em redes sociais [Fleiss et al. 1981]; bem como corretores textuais, como os disponibilizados pela biblioteca *Spacy* [Honnibal et al. 2020]. O julgamento da necessidade do uso de técnicas adicionais para o pré-processamento, assim como para qualquer outra técnica apresentada, depende da avaliação de sua necessidade dentro do contexto empregado.

2.4. Representação de Textos Utilizando Vetores Numéricos

Como mencionado na seção anterior, os dados textuais devem ser previamente pré-processados antes de poderem ser utilizados por algum modelo de linguagem. Esse procedimento, no entanto, não gera representações que podem ser compreendidas por um modelo computacional. Tais modelos conseguem lidar apenas com dados numéricos e não com textos em seu formato original. Nesse sentido, uma das formas mais simples de criar uma representação numérica é por meio do chamado *one-hot encoding*, que nada mais é do que representar um texto por meio de uma matriz de valores binários, em que cada linha representa a ocorrência de um *token* no texto.

O método consiste em usar uma representação binária única de tamanho N para cada *token* no vocabulário, sendo N o tamanho do vocabulário. Dessa forma, os documentos são representados por uma matriz de N colunas por M linhas, representando a ocorrência de cada *token* no texto. Por exemplo, um vocabulário com três palavras "está", "tudo" e "bem", poderia ser representado pelos vetores $[1,0,0]$, $[0,1,0]$ e $[0,0,1]$, respectivamente. Dessa forma, a frase "está bem" seria representada pela matriz $[[1,0,0], [0,0,1]]$, enquanto a frase "tudo bem" seria representada pela matriz $[[0,1,0], [0,0,1]]$.

Apesar da simplicidade, existem duas grandes desvantagens em se utilizar tal re-

presentação: (*i*) os vetores formados são esparsos, sendo que se o vocabulário contiver *N* *tokens*, cada vetor terá *N* dimensões com apenas uma posição com valor 1, demandando uso intensivo de memória; (*ii*) os vetores numéricos resultantes não consideram características das palavras que podem ser importantes para o domínio, por exemplo, a relevância da palavra em relação às outras de acordo com sua incidência no *corpus*, a similaridade de uma palavra com outras no *corpus*, ou a relação de uma palavra com seu contexto, o que pode impactar negativamente na capacidade do modelo de linguagem em identificar regularidades léxicas e/ou semânticas presentes nos textos.

Por isso, nesta seção é introduzido o conceito de *Embeddings*, um tipo de função parametrizada usada para mapear textos em vetores de ponto flutuante de baixa dimensionalidade (ou, do termo em inglês, *low-dimensional floating-point vectors*), que resulta em representações mais poderosas, por serem significativamente mais compactas e preservarem o relacionamento léxico (também chamado de relacionamento geométrico) e/ou relacionamento semântico entre os vetores das palavras. Desta maneira, textos semelhantes também possuem representações semelhantes no espaço vetorial de *embeddings*, sendo possível, por exemplo, mensurar a similaridade entre textos distintos através da similaridade entre seus vetores.

2.4.1. *Bag of Words* (BoW)

A noção de *Bag-of-Words* (BoW) (em tradução livre, "saco de palavras") foi introduzida por Zellig Harris [Harris 1954], e trata-se da forma mais simples de representação de palavras para um algoritmo de aprendizado de máquina. No BoW, cada documento é representado por um vetor de tamanho *N*, onde *N* é a quantidade de *tokens* distintos no vocabulário. Para manter a consistência entre diversos documentos, é necessário obter o vocabulário completo, ocorrido em todos eles. Desta forma, cada *token* único é representado por uma posição no vetor, a ser preenchida com a quantidade de vezes que o *token* ocorre no documento. Caso não haja nenhuma ocorrência de um *token* no documento, sua respectiva posição recebe o valor 0.

A vantagem do BoW é a sua facilidade de implementação, porém, há uma série de desvantagens: trata-se de um modelo que pode apresentar um grande consumo de memória para vocabulários muito extensos, gerando representações esparsas e apresentando enviesamento em relação a termos muito frequentes. Como alternativas para solução do primeiro problema, podem ser aplicados algoritmos de redução de dimensionalidade como o PCA (acrônimo de *Principal Component Analysis*) [Tan et al. 2018] e, para o problema de desbalanceamento de algumas palavras no vocabulário, temos o TF-IDF, que será apresentado a seguir.

2.4.2. *TF-IDF*: Term Frequency-Inverse Document Frequency

A intuição por trás do modelo conhecido como *TF-IDF*, proposto em [Sparck Jones 1972], está em considerar que, para um conjunto de documentos textuais sendo analisados por um método computacional, palavras que ocorrem em muitos destes documentos provavelmente não serão relevantes para distinguir o conteúdo de cada um deles [Robertson 2004]. O *TF-IDF*, portanto, é um modelo que tem como principal característica ponderar a frequência com que um determinado termo ocorra em um conjunto de documentos, em relação à quantidade de vezes em que ocorre

em um documento em específico. Desta maneira, espera-se obter uma representação balanceada e ao mesmo tempo mais precisa em relação à relevância dos termos para cada documento, atribuindo um maior peso a termos que ocorram em poucos documentos [Jurafsky and Martin 2021].

O índice IDF é computado pela Equação 1, onde N corresponde ao número de documentos analisados e df_i corresponde ao número de documentos em que ocorre o termo em questão:

$$idf = log(\frac{N}{df_i}) \tag{1}$$

Multiplicando a frequência absoluta de cada termo (*TF*) por esta ponderação, podemos representar cada elemento de uma matriz termo $\times$ documento pela Equação 2:

$$w_{i,j} = tf_{i,j} idf_i \tag{2}$$

O *TF-IDF* representou um passo importante para o desenvolvimento de técnicas relacionadas à extração de informação, tendo depois se expandido para outras técnicas da área de processamento de linguagem natural, como extração de tópicos e classificação de texto. No entanto, conforme citado anteriormente, o TF-IDF tem como principal desvantagem o fato da representação vetorial resultante possuir o mesmo tamanho do vocabulário do texto, acarretando no mesmo problema de esparsidade encontrada nas codificações do tipo *one-hot encoding* e BoW. Tal problema pode ser minimizado com a aplicação de métodos de redução dimensional como o PCA. Outra limitação para aplicá-lo em tarefas mais complexas de NLP está no fato de que se trata de uma medida puramente estatística, baseada na frequência dos termos, que não considera as proximidades semânticas em que os termos ocorrem, ou seja, não verifica as palavras que ocorrem mais próximas ou mais distantes entre si. Métodos mais recentes de *word embeddings*, comentados nas próximas subseções, buscam resolver essa segunda limitação.

2.4.3. *Word Embeddings*

A seguir, são descritos os principais métodos utilizados para geração de vetores densos (*Embeddings*), gerados a partir de modelos baseados em redes neurais. São discutidos tanto os métodos de *Word Embeddings*, que geram uma representação vetorial para cada palavra, quanto os métodos de *Sentence Embeddings* (Seção 2.4.4), que, por sua vez, consideram porções mais longas de textos, como as frases, sentenças e parágrafos.

2.4.3.1. *Word2Vec*

Com o *Word2Vec*, proposto por [Mikolov et al. 2013] inaugurou-se um novo paradigma de representação semântica vetorial. Este tipo de representação tem como principal característica a atribuição de um vetor *denso*, de tamanho arbitrário, a cada palavra de um *corpus*, gerado a partir do treinamento de redes neurais. Esses vetores são gerados a partir da análise das *janelas semânticas* em que tais palavras venham a ocorrer. Este tipo de representação trouxe uma série de inovações e vantagens: os vetores não precisam

ser treinados apenas no *corpus* em que será feita a análise (é uma prática comum que se tomem vetores pré-treinados sobre um *corpus* com vocabulário mais extenso, como a *Wikipedia*, e apenas sejam otimizados sobre o *corpus avaliado*); além disso, é eliminado o problema da esparsidade de dados, obtendo uma representação vetorial mais rica. Resultados experimentais indicam que relações semânticas complexas podem ser capturadas, como a relação entre os nomes de países e suas respectivas capitais. Além disso, sinônimos obtém representações mais próximas, enquanto antônimos se distanciam de forma equivalente no espaço vetorial [Mikolov et al. 2013, Jurafsky and Martin 2021].

O *Word2Vec*, em si, é o nome dado a dois modelos conhecidos como *Skip-Gram* e *Continuous Bag of Words* (CBOW). No modelo *Word2Vec Skip-gram*, toma-se como entrada uma palavra, e a partir dela, tenta-se prever as palavras que venham a ocorrer em sua vizinhança. No modelo *Word2Vec CBOW*, toma-se como entrada uma janela de palavras, e tenta-se prever qual palavra ocorreria naquele contexto. A Figura 2.1 ilustra a arquitetura de ambos. Como podemos ver, além das camadas de entrada e saída, há apenas uma única camada de projeção, a qual chamamos de camada oculta (*hidden layer*).

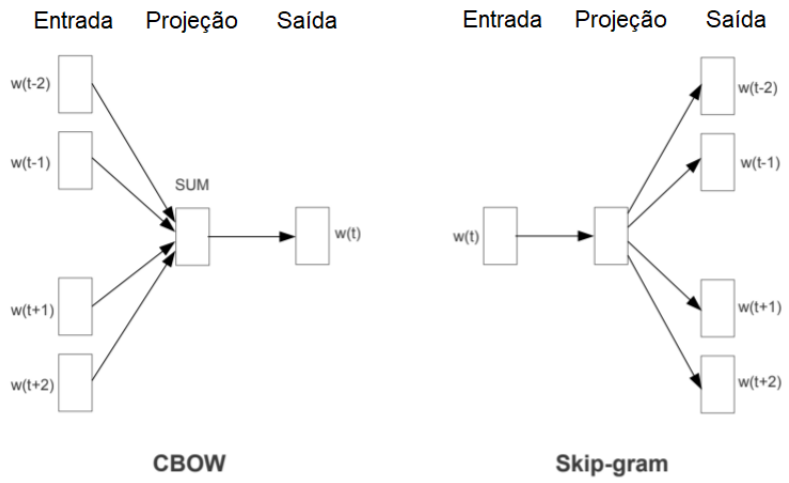


Figura 2.1. Esquema de representação dos métodos que compõem o *Word2Vec*, *CBOW* e *Skip-Gram*. Nela, o *t* indica a posição de uma palavra dentro de uma sentença. Adaptado de [Mikolov et al. 2013].

O algoritmo de aprendizado dos *embeddings* do *Word2Vec Skip-Gram* toma como entrada um *corpus* de texto, com um tamanho N de vocabulário. Inicialmente, são atribuídos valores aleatórios para cada um dos vetores das palavras do vocabulário: tais pesos são ajustados ao longo do treinamento para que palavras ocorrendo em contextos semelhantes obtenham representações vetoriais (*embeddings*) próximos, e palavras de significado distante, que não costumam ocorrer nos mesmos contextos, obtenham representações o mais distante possíveis entre si.

Este treinamento é análogo à tarefa de um classificador binário que vai recebendo instâncias *positivas* e *negativas* de treinamento: em cada palavra analisada por iteração, as instâncias positivas são palavras que realmente ocorrem em sua proximidade, e as negativas, uma seleção, de tamanho proporcional, de palavras que não ocorrem. O treinamento do algoritmo irá, então, minimizar a função de perda (*Loss Function*) representada pela

Equação 3, cujo objetivo é maximizar o produto escalar entre uma palavra e um exemplo *positivo* de contexto, e minimizar em relação aos exemplos *negativos* (as funções σ correspondem à distribuição sigmóide de probabilidade de cada um dos casos). O tamanho do contexto observado é arbitrário, sendo definido como um parâmetro de treinamento. Essa minimização é feita através de uma função de Gradiente Descendente Estocástico. Ao final, duas representações são aprendidas: uma matriz W contendo em cada vetor w_i um *word-embedding* para cada palavra do vocabulário, e uma matriz C em que cada vetor c_i é um *embedding* relativo ao contexto. [Jurafsky and Martin 2021].

$$L_{SG} = -[log\sigma(c_{pos} \cdot w) + \sum_{i=1}^{k} log\sigma(-c_{neg} \cdot w)] \tag{3}$$

Para uma descrição detalhada do treinamento do modelo da arquitetura *CBOW*, consultar [Wen et al. 2016] e [Sivakumar et al. 2020].

2.4.3.2. *Global Vectors* (GloVe)

O modelo conhecido como *GloVe* (do inglês *Global Vectors*) [Pennington et al. 2014] é uma variação de matriz de co-ocorrência, cujo objetivo é produzir vetores contínuos para representação de palavras, de maneira a minimizar o erro quadrático entre o produto vetorial entre uma palavra w e uma palavra de contexto w'. Os vetores são obtidos pela minimização da Equação 4. Nela, $X_{i,j}$ corresponde ao número de vezes em que uma palavra j aparece no contexto de uma palavra i (sendo w'_j e w_i suas representações vetoriais). X_{max} corresponde a um hiperparâmetro de valor arbitrário pré-definido (sugerido experimentalmente pelos autores do modelo como $X_{max} = 100$), e os vetores b_i e b'_j correspondem aos seus respectivos vieses:

$$J = \sum_{i,j=1}^{V} f(X_{i,j})(w_i^T w'_j + b_i + b'_j - log(X_{i,j}))^2 \tag{4}$$

O peso de cada vetor é dado por:

$$f(X_{i,j}) = \begin{cases} (X_{i,j}/X_{max})^{\alpha} & se \quad X_{i,j} < X_{max} \\ 1 & caso\ contrário. \end{cases}$$

O tamanho total dos vetores de representação de palavras, o fator escalar α (com valor experimentalmente sugerido pelos autores de $\alpha = \frac{3}{4}$) e janela de contexto, por sua vez, são arbitrários [Rendel et al. 2016].

2.4.3.3. *FastText*

Buscando solucionar uma limitação do *Word2Vec* - o fato de que este ignora a estrutura sintática das palavras, considerando cada uma delas como um *token* unitário - o modelo *FastText* propõe uma flexibilização desta representação, que permite mensurar, além da

proximidade semântica das palavras, sua proximidade léxica. Por esta razão, o *FastText* mostra-se robusto em reconhecer palavras que apresentem algum desvio em relação à grafia original (devido a erros de digitação, por exemplo), por permitir a representação de palavras fora do vocabulário original de treinamento.

No *FastText*, cada palavra é representada por uma sequência de *N*-gramas de caracteres. Por exemplo, dada uma palavra como "<farol>", onde os caracteres especiais "<" e ">" indicam o início e o final da palavra, respectivamente, e uma janela (de tamanho arbitrário, escolhido pelo usuário) $n = 3$, o *FastText* toma como entrada os 3-gramas: "<fa", "far", "aro", "rol", "ol>"; além da palavra completa "<farol>". A cada um destes *N*-gramas, é associada uma representação vetorial única. Então, a representação vetorial final da palavra, consistirá na soma dos vetores de representação de todos os seus *N*-gramas [Joulin et al. 2016, Bojanowski et al. 2017].

2.4.4. *Sentence Embeddings*

Similar a *Word Embeddings*, mas considerando porções mais longas de textos, i.e., sentenças, frases ou parágrafos, ao invés de apenas uma palavra, os métodos de *Sentence Embeddings* se destacam em relação a *Word Embeddings* por resultarem em modelos mais eficientes, ao transferir o aprendizado para outras tarefas de NLP, requerendo conjuntos de dados de treinamento menores, e obtendo desempenho superior em diversas tarefas de NLP [Cer et al. 2018]. A seguir, são descritos os principais métodos de *sentence embeddings*, desde os primeiros propostos, tais como *SkipThought*, *InferSent* e *USE*, até os modelos estado-da-arte *SBERT*, e modelos multilíngues (*cross-lingual*) como o *LASER*, o mUSE e o *LaBSE*.

2.4.4.1. SkipThought

O modelo *SkipThought* [Kiros et al. 2015], possui uma arquitetura composta por três redes neurais treinadas por meio de aprendizado não supervisionado, que geram representações vetoriais de tamanho fixo para as sentenças, sendo uma delas para codificar uma sentença de entrada e as outras duas para decodificar a sentença anterior e posterior à entrada. A premissa do modelo é de que, dada uma sentença de entrada, ele consiga identificar regularidades na sentença que permitam inferir as sentenças que vêm antes e depois dela [Kiros et al. 2015]. Por isso, durante o treinamento, cada ponto de dado é composto por três sentenças contíguas, representadas pela tripla (S_{i-1}, S_i, S_{i+1}). A camada de codificação recebe a sentença S_i, enquanto as outras duas camadas de decodificação recebem a sentença anterior (S_{i-1}) e posterior (S_{i+1}). Durante o treinamento, o erro propagado pelas camadas de decodificação é utilizado para refinar o treinamento da camada de codificação. Assim que o modelo converge, somente a camada de codificação é mantida, a qual é usada para gerar os *embeddings* de tamanho fixo, que podem ser aplicados em diferentes tarefas de NLP.

Uma das vantagens do *SkipThought* é o fato de ser um modelo não supervisionado, necessitando apenas do *corpus* de treinamento para criar o modelo de *embeddings*. Como o próprio artigo base do *SkipThought* apresenta na seção de conclusão, essa arquitetura abriu várias possibilidades para novos modelos, por exemplo, baseados em redes

convolucionais e até a expansão para trabalhar com parágrafos inteiros em vez de sentenças [Kiros et al. 2015]. Uma das limitações do *SkipThought* é a necessidade de contexto, uma vez que cada sentença depende da sentença anterior e posterior a ela em uma ordem coerente para treinar o modelo. Nesse sentido, mensagens no Twitter, por exemplo, não seriam adequadas para o treinamento desse modelo.

2.4.4.2. InferSent

O *InferSent* [Conneau et al. 2017], diferentemente do *SkipThought*, é um modelo baseado em aprendizado de máquina supervisionado, capaz de gerar representações semânticas de sentenças escritas em língua inglesa. O InferSent é treinado com o *corpus Stanford Natural Language Inference* (SNLI) [Bowman et al. 2015], que é composto por triplas contendo: (i) uma premissa (uma sentença qualquer); (ii) uma hipótese inferida a partir da premissa; e (iii) o julgamento de voluntários sobre a relação entre a premissa e a hipótese. O julgamento atribuído pelos voluntários pode assumir uma de três possíveis classes, sendo "vínculo", se a premissa possui relação com a hipótese, "contradição", se a hipótese contradiz a premissa, ou "neutro", se não é possível estabelecer uma relação entre a premissa e a hipótese.

Esse *corpus* faz parte de um campo dentro da grande área de NLP chamado *Natural Language Inference* (NLI), ou inferência de linguagem natural. A NLI compreende a tarefa de determinar se uma hipótese e respectiva premissa possuem vinculação lógica, são contraditórias, ou neutras (indeterminadas) entre si. Por exemplo, considerando a premissa "todos os dias nessa localidade são ensolarados" e a hipótese "essa localidade está nublada", a tarefa de NLI é identificar se a hipótese possui vínculo, contradição ou é neutra em relação à premissa. Conjuntos de dados usados para treinar modelos de NLI são comumente usados para treinar modelos de *embeddings*, dada a sua capacidade de generalização, como é o caso do próprio *InferSent* e do *SentenceBERT*, que será apresentado em um dos tópicos desta seção.

A arquitetura do *InferSent* possui dois codificadores idênticos para as sentenças de entrada, um para a premissa e outro para a hipótese, que geram os vetores u e v, respectivamente [Conneau et al. 2017]. Em seguida, esses vetores são concatenados para extrair a relação entre si de três formas distintas: (i) concatenando as duas representações ((u, v)); (ii) multiplicando os vetores ($u * v$); e (iii) calculando o valor absoluto da diferença entre os vetores ($|u - v|$) [Conneau et al. 2017]. A representação resultante, que captura a relação entre a premissa e a hipótese, alimenta um classificador com múltiplas camadas totalmente conectadas, além de uma camada de saída com função de ativação *Softmax* [Conneau et al. 2017]. O InferSent possui 3 saídas para representar cada uma das classes: "vínculo", "contradição" e "neutro" [Conneau et al. 2017]. Diferentes arquiteturas de redes neurais, incluindo células do tipo *Long Short-Term Memory* (LSTM) ou *Gated Recurrent Units* (GRU), e suas variações, foram testadas para os codificadores nessa arquitetura [Conneau et al. 2017]. O modelo com maior acurácia nos testes foi o que utilizou redes neurais com células LSTM bidirecionais e camada de *max pooling* (referida no artigo como BiLSTM-max) [Conneau et al. 2017]. Em testes realizados pelos autores, o modelo BiLSTM-max obteve desempenho ligeiramente superior em várias tarefas de NLP, em comparação ao *SkipThought* original, sendo um modelo capaz de generalizar

melhor em várias tarefas de *Semantic Textual Similarity* (STS), que é um campo dentro de NLP relacionado a tarefas como tradução, sumarização e geração de textos, perguntas e respostas (QA), busca semântica e sistemas conversacionais [Cer et al. 2017].

Uma vantagem da arquitetura *InferSent*, é o fato dela conseguir trabalhar com sentenças e textos com maior dimensão, além de não necessitar que o *corpus* utilizado no treinamento esteja em uma sequência lógica, como é o caso do *SkipThought*. Apesar de a arquitetura poder ser adaptada para treinamento com diferentes conjuntos de dados, a versão original treinada com os dados do SNLI está disponível apenas para língua inglesa, limitando sua aplicação a esse idioma.

2.4.4.3. *Universal Sentence Encoder* (USE)

O *Universal Sentence Encoder* (USE) [Cer et al. 2018] oferece duas maneiras para gerar *sentence embeddings* de textos escritos em inglês: (i) utilizando o subgrafo de codificação (*encoding*) da arquitetura *Transformers* [Vaswani et al. 2017]; e (ii) utilizando uma *Deep Averaging Network* (DAN) [Iyyer et al. 2015]. Em comum, ambos os métodos recebem como entrada um a sequência de palavras, em minúsculo e *tokenizada* com *Penn Treebank* (PTB) *tokenizer*, e produzem um *sentence embedding* com 512 dimensões. Apesar dessa semelhança, a forma como cada método constrói os *embeddings* das sentenças é muito distinta, como veremos a seguir.

No primeiro caso, é utilizado o mecanismo de autoatenção (*self-attention*), introduzido pela arquitetura *Transformers* [Vaswani et al. 2017]. Na arquitetura *Transformers*, o mecanismo de *self-attention* é o responsável por calcular o relacionamento entre as palavras da sentença de entrada e, enquanto ele faz isso para uma determinada palavra, ele permite que o modelo tenha foco nas outras palavras da senteça que são mais similares a esta, por isso o nome do mecanismo é atenção, por permitir que o modelo concentre-se mais no que realmente importa [Vaswani et al. 2017]. Desta forma, ao utilizar o mecanismo de *self-attention*, o modelo USE consegue computar as representações das palavras com contexto de uma dada sentença de entrada, considerando tanto a ordem como a identidade das palavras. Tais representações são convertidas em um único vetor numérico de tamanho fixo, obtido através da soma dos elementos de mesmas posições dos vetores. A principal vantagem em se utilizar esse método é o alto desempenho em diferentes tarefas de NLP, mesmo quando os conjuntos de dados possuem poucas amostras para treinamento.

Por exemplo, em [Cer et al. 2018], os autores conduziram diversos experimentos para avaliar o desempenho dos modelos em diferentes tarefas de NLP, tais como a SST (classificação binária de sentimento em nível de frase [Socher et al. 2013]) e STS *Benchmark* (similaridade textual semântica entre pares de frases [Cer et al. 2017]). Para isso, foi utilizado apenas *word embeddings* (sendo utilizado o *Word2Vec*), ou apenas *sentence embeddings*, ou a combinação de *word* e *sentence embeddings*, em relação ao *baseline*, que inicializa os pesos da rede de maneira aleatória e os *word embeddings* são aprendidos ao longo do treinamento com base nos dados da tarefa avaliada. Dentre os resultados apresentados, chama a atenção o desempenho do USE com o mecanismo de *self-attention* (chamado de USE_T) para a tarefa SST, onde foram avaliados o desempenho dos mode-

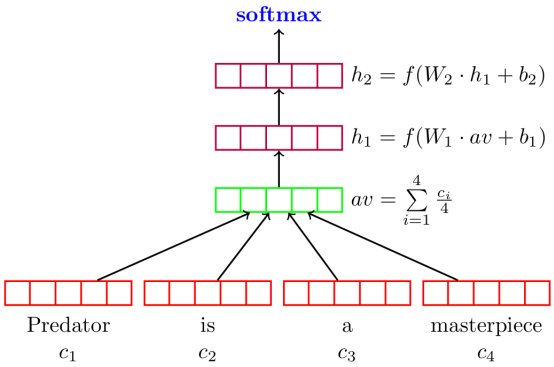


Figura 2.2. Ilustração da arquitetura do modelo DAN. Imagem de [Iyyer et al. 2015].

los considerando diferentes tamanhos do conjunto de treinamento (1.000, 2.000, 4.000, 8.000, 16.000, 32.000 e 67.300). Como mostrado, com apenas 1.000 amostras de treinamento, o modelo USE_T obteve um resultado superior em comparação com vários outros modelos, mesmo eles tendo sido treinados com o conjunto de dados de treinamento completo (i.e., 67.300 amostras). Para as demais tarefas de NLP avaliadas, os melhores resultados também foram obtidos com o USE_T, seja ele sozinho ou em conjunto com algum modelo de *word embeddings*.

Diante desses resultados, fica claro o grande potencial do USE_T para transferir seu aprendizado para muitas tarefas de NLP. No entanto, como o USE_T utiliza o *transform encoding model*, sua complexidade de tempo é $O(n^2)$, sendo N o tamanho da sentença de entrada, bem como sua complexidade de espaço também é quadrática, ou seja, $O(n^2)$. Desta forma, caso seja necessário ter uma melhor eficiência em termos de tempo, memória, ou ambos, sem degradar significativamente o desempenho, você pode optar pelo modelo USE com o *encoding* gerado com a DAN, chamado USE_D, já que ele tem complexidade de tempo linear, $O(n)$, e sua complexidade de espaço é constante no tamanho da entrada, $O(1)$.

Para combinar os múltiplos *word embeddings* das palavras de uma dada sentença de entrada em um único vetor (i.e., o *sentence embedding*), a DAN primeiro computa a média dos *word embeddings* e, então, utiliza o vetor resultante para alimentar uma ou mais camadas de uma rede neural profunda (DNN, do Inglês *Deep Neural Network*) [Iyyer et al. 2015]. A Figura 2.2, apresenta um exemplo do processo da DAN para uma frase contendo 4 palavras, utilizando duas camadas ocultas (i.e., *hidden layers*) e a classificação (linear) na última camada (i.e., *softmax*). Note que ao calcular a média dos *word embeddings* para utilizar como entrada da DNN, a informação sintática, i.e., a posição em que as palavras aparecem na frase, se perde e, por isso, a DAN é considerada uma função de composição não ordenada. Apesar disso, como é possível observar nos resultados apresentados em [Cer et al. 2017], o USE_D apresenta bom desempenho em tarefas de classificação de texto.

2.4.4.4. *SentenceBERT* (SBERT)

O *SentenceBERT* (SBERT) [Reimers and Gurevych 2019], é um modelo estado-da-arte baseado no *Bidirectional Encoder Representations from Transformers* (BERT) [Devlin et al. 2018] – um modelo considerado "divisor de águas" na área de NLP. O SBERT permite a geração de *embeddings* com mesma capacidade de generalização, mas com complexidade computacional ordens de magnitude menor em comparação com seu predecessor e outros modelos até então estado-da-arte, como o *InferSent* e USE [Reimers and Gurevych 2019]. Por isso, para entender essas vantagens, primeiramente é necessário compreender a arquitetura de seu modelo predecessor [Reimers and Gurevych 2019].

O BERT é um modelo de aprendizado profundo criado para ser facilmente adaptado a diferentes tarefas de similaridade semântica sem necessitar grandes alterações em sua estrutura (agnóstico de tarefa) [Devlin et al. 2018]. Esse modelo gera representações usando uma lógica similar à aplicada pelo *SkipThought*, onde o contexto anterior e posterior de uma palavra ou sentença é usado para identificar as características que melhor representam sua relação com seu entorno. Para isso, o BERT usa o mecanismo de atenção introduzido pela arquitetura *Transformers* [Vaswani et al. 2017]. Sua arquitetura é composta por codificadores com redes neurais de aprendizado profundo, treinados percorrendo o *corpus* tanto na ordem natural em que as sentenças ocorrem, quanto na ordem inversa, por isso o uso do termo "bidirecional" [Zhang et al. 2021]. O fato do aprendizado ocorrer em ambas as direções, permite que esse modelo possa ser pré-treinado mais rapidamente e com uma quantidade menor de dados em comparação com modelos estado-da-arte que usam aprendizado unidirecional, como o *Generative Pre-trained Transformer* (GPT) [Radford et al. 2018, Zhang et al. 2021].

A arquitetura do SBERT, por sua vez, se assemelha a do *InferSent*, contendo dois codificadores BERT com uma estrutura de redes siamesas - compartilhando os pesos entre si e necessitando apenas uma entrada - esse é o diferencial que torna esse modelo muito mais rápido em comparação ao BERT [Reimers and Gurevych 2019]. Na saída de cada um dos codificadores BERT, uma camada de *pooling* é responsável por gerar duas representações vetoriais, u e v, de tamanhos fixos [Reimers and Gurevych 2019]. Para tarefas de classificação, os vetores u e v, então, são transformados em uma tripla contendo (i) u; (ii) v; e (iii) o valor absoluto da subtração entre o vetor u e v ($|u - v|$) [Reimers and Gurevych 2019]. Essa tripla é passada para um modelo de classificação usando a função de ativação *Softmax* [Reimers and Gurevych 2019]. Já para tarefas de regressão, a similaridade do cosseno entre os vetores u e v é computada após a camada de *pooling* e enviada para a saída do modelo [Reimers and Gurevych 2019].

Com essa arquitetura, o SBERT, além de ter apresentado melhor desempenho em relação ao *InferSent* e USE, também resolve os problemas de complexidade computacional apresentados pelo BERT [Reimers and Gurevych 2019]. Além disso, o SBERT possui outras características que o tornam bastante atrativo, não apenas na área acadêmica, mas em aplicações na indústria. Por exemplo, modelos SBERT monolíngues podem ser aumentados para tarefas de NLP multilíngues com boa capacidade de generalização, partindo da premissa de que uma sentença traduzida deve ser mapeada para a mesma posição no espaço vetorial que a sentença original [Reimers and Gurevych 2020]. Esse é o pro-

cesso usado por um famoso modelo de análise de toxicidade em comentários na *web* chamado Perspective API [Jigsaw 2022], que será apresentado em uma das seções desse capítulo.

Outra vantagem do SBERT é o fato deste modelo estar disponível por meio de uma biblioteca de código aberto feita em Python, chamada *Sentence-Transformes*[12], que disponibiliza vários modelos pré-treinados para diferentes tarefas, incluindo modelos multilíngues. O projeto tem uma documentação bem estruturada e expõe alguns métodos fáceis de usar para a geração de *embeddings*, bastando apenas indicar o nome do modelo pré-treinado que será usado e aplicá-lo para gerar os *embeddings*. A transferência de aprendizado também pode ser facilmente feita com essa biblioteca, permitindo adaptar qualquer modelo disponibilizado a diferentes tarefas de NLP.

2.4.4.5. Modelos Multilíngues

Modelos multilíngues, também conhecidos como independente de idioma (do termo em Inglês *language-agnostic*), têm a capacidade de gerar *embeddings* similares para palavras com o mesmo significado mas escritas em idiomas diferentes. Ou seja, no espaço vetorial de *embeddings*, as palavras bom, *good*, *bueno*, *bien*, *gut* e *bene*, terão representações vetoriais muito semelhates, caso o modelo multilíngue suporte seus respectivos idiomas, Português, Inglês, Espanhol, Francês, Alemão e Italiano.

Note que essa caraterística é muito interessante, uma vez que pode ser muito difícil ter *corpus* disponíveis em diferentes idiomas para treinar os modelos de linguagem, sendo a maioria deles escritos em língua Inglesa. Além disso, ao lidar com textos de mídias sociais, é muito comum se deparar com textos escritos em outros idiomas. Por exemplo, ao coletar *tweets* compartilhados por pessoas que estão em uma cidade cosmopolita como Nova Iorque, EUA, além de textos escritos em Inglês, como é esperado, também é comum obter *tweets* escritos em Português, Espanhol, Mandarim, entre outros idiomas. Desta forma, utilizar modelos multilíngues para geração dos *embeddings* podem ser uma boa opção.

Como vimos na seção anterior, o SBERT pode ser treinado para tarefas de NLP multilíngues. Além dele, são apresentados a seguir outros três modelos multílingues amplamente utilizados pela academia e indústria.

- *Language-Agnostic SEntence Representations* (LASER) [Artetxe and Schwenk 2019], foi o primeiro modelo a explorar a representação de sentenças multilíngues para propósito geral, ou seja, sem ter uma tarefa de NLP específica. Para isso, o LASER utiliza uma arquitetura *sequence-to-sequence encoder-decoder*, como mostra a Figura 2.3. Como podemos ver, o *encoder* é composto por uma rede LSTM bidirecional (ou simplesmente, BiLSTM), que recebe como entrada uma sequência de palavras (representadas pelas variáveis $x_1, x_2, ...$), seguida do símbolo que indica o final da sentença (i.e., $</s>$). Cada palavra é primeiro codificada por um vocabulário de *Byte-Pair encoding* (BPE), que foi construído considerando a concatenação de todos os corpora de treinamento

[12]https://www.sbert.net. Último acesso em 11 de Setembro de 2022.

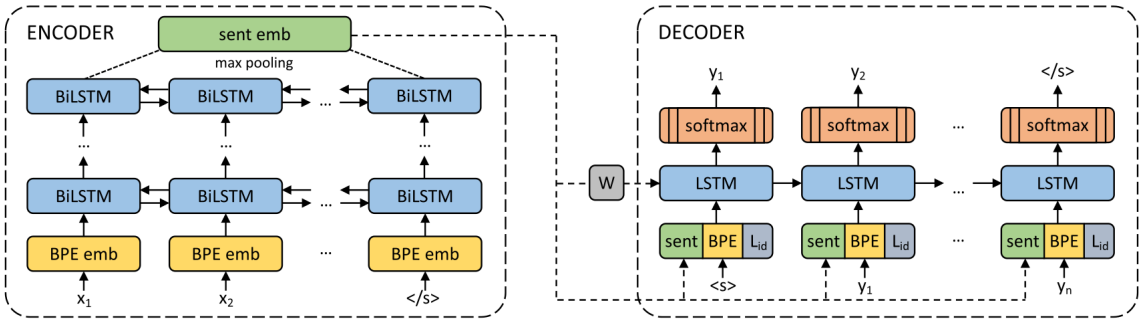


Figura 2.3. Ilustração da arquitetura do modelo LASER. Imagem de [Iyyer et al. 2015].

dos 93 idiomas suportados pelo modelo, antes de alimentar a rede BiLSTM. Com isso, não é necessário informar explicitamente o idioma de entrada para o *encoder*, o que auxília o modelo a aprender as representações independente do idioma. Por fim, o *sentence embedding* é obtido no *encoder* ao aplicar a operação *max pooling* sobre a BiLSTM. O *decoder* por sua vez, que é composto por apenas uma rede LSTM, recebe como entrada o *sentence embedding* gerado pelo *encoder*, concatenado em cada etapa do tempo com a codificação BPE da sentença alvo (i.e., as palavras $<s>, y_1, ..., y_n$) e a codificação que específica qual idioma deve ser gerado (representado pela variável L_{id}).

Assim, considerando dados anotados em apenas dois idiomas alvos, onde foram utilizados Inglês e Espanhol, o LASER foi treinado de maneira *end-to-end* para tarefa de tradução. Após o treinamento, o *encoder* torna-se capaz de gerar *sentence embeddings* em qualquer um dos 93 idiomas do conjunto de treinamento, de modo que sentenças semelhantes em diferentes idiomas também estejam próximas no espaço de *embeddings*.

- *Multilingual Universal Sentence Encoder* (mUSE) [Yang et al. 2019] é uma extensão do modelo USE [Cer et al. 2018] (descrito na Seção 2.4.4.3), onde são adicionados dois modelos multilíngues multitarefas, sendo um baseado em Rede Neural Convolucional (*Convolutional Neural Network*, CNN) e outro na arquitetura *Transformers*, além de um modelo *Transformers* multilíngue para uso em recuperação de perguntas e respostas (*Retrieval Question Answering*, ReQA). Em comum, os três novos modelos são treinados utilizando a abordagem *multi-task dual-encoder* [Chidambaram et al. 2018], que é capaz de aprender representações semelhantes para frases com significados idênticos, pórem, escritas em idiomas distintos, por meio de *bridging translation task* [Chidambaram et al. 2018]. Ao todo, 16 idiomas distintos são suportados pelo mUSE.

- *Language-agnostic BERT Sentence Embedding* (LaBSE) [Feng et al. 2020], também utiliza a abordagem *dual-encoders*, semelhante ao mUSE, para aprender representações multilíngues. Nesta abordagem, pares de sentenças de origem e alvo são codificadas separadamente, onde os *embeddings* de cada uma são obtidos por *encoders* distintos, mas que compartilham seus parâmetros. Os *embeddings* são então treinados considerando a tarefa de tradução. Como podemos ver na Figura 2.4,

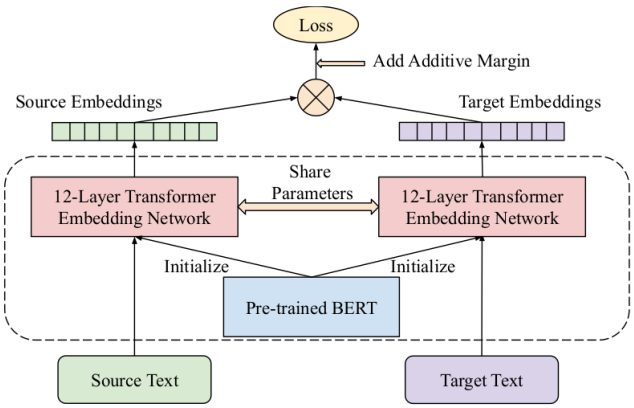


Figura 2.4. Ilustração da arquitetura do modelo LaBSE. Imagem de [Feng et al. 2020].

o principal diferencial do LaBSE, em relação aos demais modelos multilíngues que utilizam essa abordagem, é a combinação de um *encoder* pré-treinado baseado no modelo de linguagem BERT com os *dual-encoders*, evitando assim a necessidade de treiná-los do zero. Com isso, o LaBSE consegue reduzir drasticamente a quantidade de treinamento necessário e, ainda assim, alcançar um desempenho superior. Ao todo, 109 idiomas distintos são suportados pelo LaBSE. Além disso, o LaBSE também foi capaz de produzir bons resultados para mais de 30 idiomas além dos que já são suportados, mesmo não havendo quaisquer dados de treinamento de tais idiomas.

2.5. Modelagem e Extração de Conhecimento

Nesta seção, serão apresentadas técnicas de modelagem que podem ser aplicadas para extrair conhecimento a partir de representações vetoriais de textos. Dentre os tópicos que serão apresentados, estão as técnicas para agrupamento de textos similares, incluindo o agrupamento com *k-means*, agrupamento hierárquico e agrupamento usando algoritmos de detecção de comunidades em grafos. Por último, serão apresentadas técnicas para modelagem de tópicos e seus respectivos desafios.

2.5.1. Agrupamento de Textos

O problema da clusterização ou agrupamento, pode ser definido como o de identificar grupos de objetos similares em um conjunto de dados [Aggarwal and Zhai 2012]. Em NLP, o agrupamento de textos pode ser aplicado em diferentes granularidades, por exemplo, a nível de documento, parágrafos, sentenças ou termos [Aggarwal and Zhai 2012]. Além do simples agrupamento de textos, também é possível realizar a organização hierárquica, sumarização e classificação de documentos – no último caso, transformando a tarefa de clusterização em uma tarefa de aprendizado supervisionado [Aggarwal and Zhai 2012]. Várias técnicas podem ser utilizadas para realizar clusterização, incluindo abordagens clássicas, como o *k-means* e o agrupamento hierárquico, assim como a transformação de textos em grafos para aplicação de algoritmos de detecção de comunidades.

2.5.1.1. *k*-means

Um dos mais famosos métodos de agrupamento é o *k-means*, dado seu funcionamento simples e intuitivo. Inicialmente, deve-se escolher o número de agrupamentos (ou, *clusters*), em que o conjunto de documentos deve ser particionado, dado pelo parâmetro k, que pode ser definido com o auxílio de heurísticas [Tan et al. 2018]. Então, são definidos os centróides iniciais para cada um dos k *clusters* em um espaço multidimensional, podendo inicialmente serem aleatórios ou definidos por alguma heurística [MacQueen 1967]. Depois, é calculada a similaridade da representação vetorial de cada documento com todos os centróides dos *clusters* usando, geralmente, a distância Euclidiana ou alguma outra métrica de similaridade [MacQueen 1967]. Em seguida, cada documento é alocado ao *cluster* com maior similaridade com seu centróide [MacQueen 1967]. Feito isso, o centróide de cada *cluster* é atualizado pela média da representação vetorial dos documentos no *cluster* (por isso o nome *k-means*) e repetido o processo de cálculo de similidade até que o algoritmo atinja um critério de parada (i.e., um número máximo pré-definido de iterações ou quando não houver alterações significativas nos centróides) [MacQueen 1967].

Esse método pode ser aplicado sobre representações vetoriais ingênuas de documentos, como o *one-hot-encoding*, apresentando resultados satisfatórios para algumas aplicações. Contudo, representações simples como essa, em geral, resultam em vetores esparsos que dificultam o cálculo da similaridade entre documentos. Esse é o caso, por exemplo, de textos curtos em um *corpus* com vocabulário extenso, como no caso de mensagens do Twitter, ou comentários em postagens de mídias sociais em geral, onde a complexidade no vocabulário é maior devido à presença de gírias, erros gramaticais e linguagem da Internet. Por isso, representações vetoriais mais sofisticadas, como *word/sentence embeddings*, que são capazes de capturar melhor as regularidades léxicas e/ou semânticas presentes nos textos, são preferíveis nesse tipo de aplicação [Aggarwal and Zhai 2012].

Em análise de textos de mídias sociais, a velocidade com que um algoritmo consegue produzir um resultado pode ser determinante para sua escolha, uma vez que nesses ambientes digitais são produzidos grandes volumes de dados para serem processados, sendo comum algumas aplicações demandarem que isso seja feito, inclusive, em tempo real. Nesse sentido, o algoritmo *k-means* original tem um custo computacional alto de $O(kns)$, onde n é o número de documentos e s é o tamanho da representação vetorial de cada documento [Sculley 2010]. Uma forma de reduzir o custo computacional é removendo termos pouco representativos no *corpus*, consequentemente reduzindo o valor s, por exemplo, usando o próprio valor de TF-IDF para filtrar termos muito comuns ou muito raros no vocabulário. Apesar disso, ainda assim, essa redução pode não ser suficiente para grandes conjuntos de dados. Por isso, foram propostas diversas variações do algoritmo *k-means*, uma delas, dentre as mais famosas, chamada *mini-batch k-means* [Sculley 2010]. Essa abordagem trabalha com amostragens aleatórias de documentos para serem comparados com os centróides, dado por um parâmetro, que torna o algoritmo ordens de magnitude mais rápido que a versão original, possibilitando clusterização de documentos praticamente em tempo real em diversas aplicações [Sculley 2010].

2.5.1.2. Agrupamento Hierárquico

Um dos desafios em grandes *corpus* de domínios do mundo real, como no caso de textos de mídias sociais, é não saber de antemão informações quantitativas sobre como documentos estão organizados [Ushioda 1996]. O agrupamento com *k-means*, como apresentado na seção anterior, colabora nessa tarefa. Contudo, em algumas aplicações, pode ser difícil escolher o valor de k e compreender o motivo de um determinado agrupamento ter sido identificado, principalmente em grandes *corpus*. Para minimizar esse problema, existe uma classe especial de métodos de agrupamento, chamada agrupamento hierárquico. O agrupamento hierárquico permite agrupar as representações dos textos em diferentes níveis de granularidade, criando organizações dos *clusters* e *subclusters* na forma de dendrogramas. Essa é uma representação conveniente, que ajuda a responder perguntas como "Quantos grupos úteis existem nestes dados?" e "Quais são as inter-relações mais importantes?" [Murtagh and Contreras 2012].

Existem dois métodos para realizar o agrupamento hierárquico: (i) por meio de algoritmos divisivos, e (ii) por meio de algoritmos aglomerativos. Algoritmos hierárquicos divisivos [Zhao and Karypis 2002], iniciam o processo de particionamento dividindo o conjunto de documentos inicial em dois *clusters*. Em seguida, os *clusters* formados (contendo mais de um documento), são divididos ao meio, baseando-se em uma heurística dada por uma função de critério de clusterização. Esse processo continua $n-1$ vezes, sendo n o número de documentos. Ao final, cada *cluster* irá conter apenas um documento, chamado de nós folhas. Nessa abordagem, o dendrograma é construído de cima para baixo, partindo de um único *cluster* com todos os documentos e terminando com cada documento em seu próprio *cluster* [Zhao and Karypis 2002]. De maneira inversa, os algoritmos hierárquicos aglomerativos [Zhao and Karypis 2002], começam com um documento em cada *cluster* e, progressivamente, une os pares de *clusters* entre si, até resultar em um único *cluster* com todos os documentos inclusos.

Os algoritmos de agrupamento divisivo, em geral, definem o problema de agrupamento como o cálculo de uma solução de agrupamento tal que o valor de uma função de critério de clusterização seja otimizado [Zhao and Karypis 2002]. As funções de critério de clusterização podem ser dividas em 4 grupos: (a) internas, (b) externas, (c) híbridas e (d) baseadas em grafos. As funções internas (a), consideram na similaridade entre os documentos do *cluster* a ser particionado, desconsiderando documentos fora dele; as funções externas (b), se baseiam em quanto o *cluster* que será particionado se difere de outros *clusters*; as funções baseadas em grafos (c), modelam os documentos como grafos e usam métricas de clusterização; e as funções híbridas (d) otimizam múltiplas funções de critério de clusterização [Zhao and Karypis 2002].

Os algoritmos de agrupamento aglomerativo, por sua vez, podem aplicar diferentes métodos para identificação dos pares de *clusters*, a serem agrupados em cada passo [Tan et al. 2018]. Existem vários métodos feitos para aplicações específicas, dentre eles, o de *linkagem* única e o de *linkagem* completa. O esquema de *linkagem* única, agrupa dois *clusters* que contêm o par de documentos com a menor distância entre si, dentre todos os pares de documentos no *corpus* [Omran et al. 2007]. Já o esquema de *linkagem* completa, agrupa dois *clusters* quando a distância entre o par de documentos mais distantes entre si for a menor dentre todos os pares de documentos no *corpus* [Omran et al. 2007]. Há

também um método muito similar ao aplicado pelo *k-means*, chamado método centróide, onde o agrupamento é feito entre dois *clusters* que tenham a menor distância entre seus pontos médios (centróides) [Tan et al. 2018].

Em relação à complexidade computacional, os algoritmos de agrupamento hierárquico aglomerativo possuem dois passos que aumentam seu custo computacional, sendo o primeiro deles a comparação da similaridade entre os pares de todos os documentos, repetindo esse procedimento em cada passo ao subir na árvore de resultados. Por isso, esses algoritmos possuem complexidade polinomial, geralmente de $O(n^2)$ [Zhao and Karypis 2002]. Já os algoritmos de agrupamento divisivos, dada a natureza de "dividir para conquistar", têm complexidade de $O(nlog(n))$ ou, em muitos casos, linear – quantos mais balanceados os *clusters*, mais rapidamente o algoritmo converge [Zhao and Karypis 2002].

2.5.1.3. Detecção de Comunidades em Grafos

Grafos no contexto de NLP podem ser interessantes pelo poder de expressarem relações complexas entre as entidades. Comunidades, também chamadas *clusters* ou agrupamentos em grafos, são estruturas formadas por vértices que apresentam características em comum. No contexto de NLP, a detecção de comunidades pode ser utilizada para identificar grupos de textos similares. Para isso, o primeiro passo consiste em modelar um grafo que representa um dado *corpus*. Isso pode ser feito de diferentes formas, por exemplo, construir um grafo ponderado e não direcionado, onde os vértices representam os *tokens* presentes no *corpus* (i.e., o vocabulário), as arestas indicam alguma relação entre os *tokens* e, o peso das arestas, reflentem a intensidade dessa relação.

Para fins ilustrativos dessas ideias, considere o grafo construído em [Santos et al. 2020a]. Nesse trabalho, os autores primeiro realizaram *Part-Of-Speech* (POS) *tagging* em todas as sentenças de um *corpus*, o qual é composto por aproximadamente 40 mil comentários de usuários das mídias sociais Google Places e Foursquare (Tips). Com isso, foi possível identificar quais *tokens* eram um adjetivo, por ser a classe de palavras que designa qualidade e, por isso, foi a escolhida pelos autores. Ao todo, foram identicados 271 *tokens* como sendo adjetivos. O próximo passo, foi identificar a similaridade sintática e semântica entre esses *tokens*, para ponderar as arestas de um grafo. Para isso, o mesmo *corpus* foi utilizado para treinar o modelo Word2Vec [Mikolov et al. 2013] e, assim, ser possível computar a similaridade entre os 271 *tokens*. Adicionalmente, também foi considerado a polaridade do sentimento (podendo ser positiva, negativa ou neutra, como é discutido em mais detalhes na Seção 2.6.3), para compor a pontuação final de similaridade entre os *tokens*. A utilização da polaridade do sentimento nesse caso é bastante relevante, para aumentar a pontuação entre *tokens* de mesma polaridade e, por outro lado, penalizar quando os *tokens* têm polaridades opostas. Isto é devido a palavras como "ótimo" e "péssimo", ou "alegre" e "triste", que podem ter contextos muito parecidos, mas sentidos opostos. Por exemplo, "meu dia está sendo ótimo" e "meu dia está sendo péssimo", ou, "recebi uma notícia que alegrou o meu dia" e "recebi uma notícia que entristeceu o meu dia". Em ambos os exemplos, as expressões são quase iguais, exceto pelos adjetivos.

De posse dos *tokens* e das pontuações de similaridade, os autores construíram um grafo não-direcionado, completo e ponderado, denotado por $K_n = (V, E, \omega)$, onde V é o conjunto de vértices que representa os 271 *tokens* (i.e., $|V| = n = 271$), E é o conjunto de arestas entre todos os vértices, onde cada aresta $e \in E$ tem o peso $\omega_e \in \omega$, definido pela pontuação de similaridade. Antes de realizar a detecção de comunidades no grafo K_n, os autores removeram as arestas consideradas "leves", aquelas cuja a pontuação é menor que um certo *threshold*, afim de remover conexões entre vértices pouco similares. Então, após a remoção de tais arestas, os vértices que ficaram isolados também foram removidos, resultando em um grafo com 261 vértices e 3485 arestas, que foi utilizado para detecção de comunidades.

O método escolhido pelos autores para detecção de comunidades foi o *clique percolation* [Palla et al. 2005], que identifica k-clique comunidades como sendo a união de todas as cliques de tamanho k que podem ser alcançadas através de cliques adjacentes (i.e., que compartilham $k-1$ vértices). Como resultado, em [Santos et al. 2020a], foram identificadas 8 comunidades, considerando $k = 6$.

O método *clique percolation* é particularmente interessante, porque possibilita a remoção de possíveis ruídos, uma vez que vértices fracamente conectados não são colocados em nenhuma comunidade. Além disso, também permite a existência de comunidades sobrepostas, ou seja, um mesmo vértice pode pertencer a várias comunidades e, assim, evita a divisão de grandes comunidades em pequenas, mantendo comunidades com contextos semelhantes unidas. No entando, este método possuí duas principais desvantagens: definir o valor adequado de k; e, como o problema k-clique é $\mathscr{NP}$-difícil, no pior caso, sua complexidade é exponencial no número de vértices.

Alternativamente, outro método bastante conhecido para detecção de comunidades em grafos é o *Label Propagation Algorithm* (LPA) [Zhu and Ghahramani 2002] que, diferente do método *clique percolation*, é rápido (complexidade de tempo quase linear no tamanho da entrada – o grafo) e considera apenas a estrutura do grafo para identificar as comuninades, sem necessitar definir a priori nenhuma função objetivo ou parâmetro. Para isso, o LPA considera inicialmente que cada objeto é uma comunidade (ou *cluster*). Então, de maneira interativa, o LPA atualiza a *label* de cada vértice do grafo, de acordo com a *label* majoritária entre os seus vizinhos e, em caso de empate entre duas ou mais *labels*, uma é escolhida de maneira aleatória. Quando o algoritmo converge, i.e., cada nó já possui a *label* majoritária de seus vizinhos, então ele se encerra e os vértices que terminarem esse processo com a mesma *label* são considerados como sendo da mesma comunidade. Também é possível definir um número máximo de iterações, para encerrar o processo antes da convergência. Além disso, é possível utilizar o LPA de maneira semi-supervisionada, onde se atribuí algumas *labels* preliminarmente, i.e., algumas comunidades são definidas no início, restringindo assim as possíveis soluções resultantes.

2.5.2. Modelagem de Tópicos

Um objetivo comum em NLP é identificar estruturas semânticas compartilhadas entre textos para determinar quais eventos, conceitos ou assuntos estão sendo discutidos em um conjunto de documentos [Vayansky and Kumar 2020]. Esse é o papel da modelagem de tópicos, um processo que envolve a aplicação de métodos estatísticos para descoberta de estruturas semânticas latentes, também chamadas de tópicos, em um conjunto de textos

[Huh and Fienberg 2012]. Para isso, parte-se da premissa de que cada documento pode ser representado por uma mistura de tópicos, sendo cada tópico composto por palavras que melhor o define no *corpus* analisado [Huh and Fienberg 2012]. Por exemplo, em um *corpus* de notícias de várias fontes, os tópicos poderiam representar eventos que ocorreram (e.g., confronto entre Rússia e Ucrânia, Copa do Mundo, etc.) ou delinear grandes temas (e.g., economia, política, educação, etc.). A principal tarefa na modelagem de tópicos, portanto, é descobrir padrões de uso de palavras e relacionar documentos que compartilham padrões semelhantes [Alghamdi and Alfalqi 2015].

Quando aplicada a modelagem de tópicos, para a maioria dos casos, não é necessário considerar a ordem em que as palavras ocorrem nos documentos, sendo geralmente usada uma representação vetorial de *Bag of Words* (BoW) para os textos [Huh and Fienberg 2012]. De uma forma simplificada, em um modelo de tópicos cada documento pode ser representado por um histograma com a contagem de cada termo contido nele [Huh and Fienberg 2012]. A forma desse histograma é proveniente de uma distribuição entre k tópicos pré-definidos, os quais são distribuídos entre os termos no vocabulário do *corpus* [Huh and Fienberg 2012]. O objetivo da modelagem de tópicos, então, é aprender essas distribuições para conseguir inferir estruturas semânticas latentes no *corpus* [Huh and Fienberg 2012]. Para isso, as técnicas de modelagem de tópicos geralmente incluem dois componentes principais: uma matriz que relaciona os termos do vocabulário aos k tópicos e outra matriz que relaciona os k tópicos aos documentos [Huh and Fienberg 2012].

Existem várias técnicas conhecidas para modelagem de tópicos, tais como o *Latent Semantic Analysis* (LSA) [Landauer et al. 1998] – não probabilístico –, e sua respectiva versão probabilística (pLSA) [Hofmann 2001], e o *Latent Dirichlet Allocation* (LDA) [Blei et al. 2003]. Elas podem ser facilmente aplicadas por meio das implementações disponíveis na biblioteca GenSim [Řehůřek and Sojka 2010], que além da implementação de todas essas técnicas, também possui uma versão otimizada do LDA com processamento paralelizado e métricas para avaliação de desempenho. Outra biblioteca mais recente, chamada OCTIS [Terragni et al. 2021], permite a criação de modelos de tópicos sem a necessidade de otimizar os hiperparâmetros manualmente, os quais são estimados de forma automática por meio de uma abordagem Bayesiana. Essa biblioteca usa a mesma implementação do GenSim [Řehůřek and Sojka 2010], portanto, para efeitos de comparação, são exatamente as mesmas implementações. Outra vantagem da biblioteca OCTIS, é o fato dela possuir uma interface gráfica, que permite a prototipação rápida de modelos, sem a necessidade de criar código, facilitando com que pesquisadores de diferentes domínios possam criar modelos de tópicos para desafios específicos em suas áreas [Terragni et al. 2021].

Uma característica que deve ser considerada ao modelar tópicos em grandes *corpus* é sua evolução ao longo do tempo. Essa característica, em alguns casos, pode comprometer o processo de descoberta de tópicos, uma vez que os métodos tradicionais, como o LSA, pLSA e LDA, não capturam a relação entre os tópicos mais novos com seus predecessores [Alghamdi and Alfalqi 2015]. Um exemplo prático deste caso, seria o de um pesquisador cujo objetivo é identificar temas dentro de um determinado campo de pesquisa ao longo de décadas, de forma que seja possível obter conjuntos de artigos que tratam do mesmo tema, mesmo que algumas palavras-chave tenham sido alteradas ao longo do

tempo [Alghamdi and Alfalqi 2015]. Algumas técnicas conhecidas para modelagem de tópicos no decorrer do tempo são o *Topic Over Time* (TOT) [Wang and McCallum 2006] e *Dynamic Topic Models* (DTM) [Huh and Fienberg 2012]. Essas técnicas adicionam um novo componente no modelo de tópicos, que é a relação entre a metainformação de *timestamp* dos documentos com os tópicos. Dessa forma, os modelos temporais têm o objetivo adicional de aprender a distribuição dos tópicos ao longo do tempo em comparação com os modelos atemporais [Wang and McCallum 2006, Huh and Fienberg 2012].

Alguns desafios em modelagem de tópicos devem ser destacados. A caracterização de tópicos é uma tarefa complexa, uma vez que é necessário um avaliador humano para identificar qual é o assunto a partir das *top* N palavras mais representativas de um tópico [Ramage et al. 2009]. Nesse sentido, dar um nome significativo para um tópico é difícil, e depende de um profundo entendimento do domínio e vocabulário, além das nuances representadas pelas *top* N palavras mais representativas do tópico, que nem sempre delimitam de forma clara um tema específico [Ramage et al. 2009]. Em diferentes contextos, tópicos têm significados diferentes, por exemplo, em um *corpus* que evoluiu ao longo do tempo, pode acontecer de alguns tópicos inicialmente ligados entre si serem separados em algum momento, e essa dinâmica pode não ser capturada pelo modelo [Ramage et al. 2009]. Por fim, a modelagem de tópicos depende do componente humano, devendo este, preferencialmente, ser um especialista no domínio, para ser possível identificar um número razoável k de tópicos, além de ser capaz de avaliar se eles possuem relação com o que se pretende estudar [Ramage et al. 2009]. Considerando que não existe forma estritamente objetiva para escolher os k tópicos, sendo a qualidade desta tarefa diretamente relacionada à habilidade do avaliador, que deve saber de antemão quais tópicos podem existir no *corpus* e suas respectivas interpretações, aumenta a probabilidade de enviesar as análises posteriores [Ramage et al. 2009]. No entanto, existe heurísticas, como por exemplo, a medida de coerência proposta em [Röder et al. 2015], que permite avaliar a qualidade dos tópicos identificados por modelos de tópicos e, assim, auxiliar na descoberta do número de tópicos em um *corpus*.

2.6. Compreensão Semântica e Emocional

A Compreensão de Linguagem Natural, ou *Natural Language Understanding* (NLU), como o próprio nome indica, auxilia na compreensão semântica de textos em linguagem natural. Para isso, a NLU, comumente, realiza duas tarefas: detecção (ou reconhecimento) de intenção e reconhecimento de entidades nomeadas. Assim, para cada texto em linguagem natural (também chamado de expressão ou, ainda, *utterance*), a NLU retorna como resultado de seu processamento a intenção do texto, isto é, o objetivo que a pessoa tinha em mente ao enviar a expressão, e as entidades nomeadas que foram reconhecidas. Além disso, também pode-se incluir a detecção de sentimentos e emoção a esse processo para ser possível obter também a compreensão emocional. Nesta seção são apresentados mais detalhes sobre as técnicas que compõem a NLU.

2.6.1. Detecção de intenção

A detecção de intenções é a parte responsável por fornecer uma interpretação geral do significado de uma expressão. Em outras palavras, a intenção nos informa o que uma expressão quer dizer. Normalmente, é uma tarefa abstraída em um processo de classifi-

cação, na qual se treina um modelo de classificação supervisionado com um conjunto de expressões rotuladas, onde para cada expressão é atribuída uma intenção correspondente.

Para dados de mídias sociais, normalmente, não temos a disposição a intenção (podendo ser uma categoria, ou assunto, ou tema, etc) dos textos compartilhados, que poderia nos auxiliar no processo de construção do conjunto de dados para treinamento do modelo de classificação. Além disso, os usuários podem compartilhar conteúdos diversos, desde de política, esporte, cultura, religião, até situações pessoais cotidianas, como suas lembranças (conhecido como *tbt*, do termo "*Throwback Thursday*"), declarações amorosas e momentos especiais. Então, uma possível pergunta seria: quais são as possíveis intenções em *corpus* de textos de mídias sociais?

Para responder a essa questão, podemos utilizar a modelagem de tópicos (como descrito na Seção 2.5), para determinar quais assuntos estão presentes no *corpus*, como mostra [Churchill and Singh 2021, Aiello et al. 2013], onde são analisados várias abordagens para mineiração de textos de mídias sociais e identificação de tópicos. Desta maneira, após identificar os tópicos presentes no *corpus* e atribuir um assunto para cada um deles, de acordo com as palavras que o melhor representam, torna-se mais simples a criação do conjunto expressões rotuladas.

De posse do conjunto de expressões rotuladas, o próximo passo é treinar algum modelo de classificação, para que ele seja capaz de inferir a intenção mais provável de uma expressão inédita (i.e., uma expressão diferente das expressões utilizadas durante o treinamento do modelo). Para isso, existe várias possibilidades, como utilizar uma arquitetura similar ao modelo LASER [Artetxe and Schwenk 2019], sendo uma rede BiLSTM, seguida de uma rede LSTM e uma camada totalmente conectada (função ativação) para realizar a classificação, que é similar ao modelo utilizado em [Yang et al. 2017]. Ou, ainda, utilizar algum modelo de *sentence embeddings* baseado na arquitetura *Transformers* (como descrito na Seção 2.4.4), que é o estado-da-arte para várias tarefas de NLP, entre elas, a classificação de textos. Assim, pode-se utilizar o modelo LaBSE [Feng et al. 2020] para obtenção dos *embeddings*, seguido de algum algoritmo de classificação (*Random Forest*, *Logistic Regression CV*, etc). Em [Ladeira et al. 2022], são avaliadas diferentes combinações de modelos de *sentence embeddings* e algoritmos de classificação para detecção de intenções, considerando 5 conjuntos distintos de expressões rotuladas.

2.6.2. Reconhecimento de entidades nomeadas

O reconhecimento de entidades nomeadas (do inglês *Named Entity Recognition*, ou NER), corresponde à tarefa de reconhecimento e categorização de termos relevantes em um texto, para enriquecimento da interpretação das informações contidas neste texto. Esses termos relevantes são os substantivos correspondentes a nomes próprios, por exemplo, pessoas, empresas, lugares, produtos, organizações, cores, valores monetários, idioma, GPE (países, estados, cidades), dentre outros.

Desta forma, a tarefa do NER é reconhecer dentre os termos de um texto escrito em linguagem natural, quais podem ser caracterizados como nomes próprios e quais categorias de nome provavelmente se referem, o que é chamado de entidades nomeadas. Há portanto, diferentes técnicas para identificar as entidades nomeadas, por exemplo, o *slot*

filling.

No *slot filling*, cada palavra (ou termo) de um texto é marcada com um *slot*. Para isso, é utilizado algum tipo de notação, por exemplo, a popular notação *in/out/begin* (IOB), onde "B" denota o início de uma entidade, "I" significa "dentro" e é usado para todas as palavras que compõem a entidade, exceto a primeira, e "O" significa ausência de entidade.

A Tabela 2.1 mostra um exemplo de *slot filling* inspirado em [Mesnil et al. 2014], que utiliza o bem conhecido *benchmark* chamado *Airline Travel Information System* (ATIS). Como podemos ver, as palavras "Boston" e "New York" foram identificadas como sendo os locais de partida e chegada, respectivamente. Além disso, a palavra "today" foi identificada como uma data. A partir deste resultado, é possível compreender com base nos *slots* identificados, quais são as cidades de origem e destino e, também, a data da viagem.

Frase	*show*	*flights*	*from*	*Boston*	*to*	*New*	*York*	*today*
Slots	O	O	O	B-dept	O	B-arr	I-arr	B-date

Tabela 2.1. Exemplo de *slot filling* inspirado em [Mesnil et al. 2014].

Para realizar esta tarefa, a abordagem mais popular é o *Conditional Random Field* (CRF) [Lafferty et al. 2001] e suas variantes. O modelo CRF de cadeia linear, é um método de classificação discriminativa capaz de prever a sequência mais provável de rótulos (ou *slots*) para uma sequência de palavras. Recentemente, as abordagens de aprendizado profundo se tornaram mais populares para preenchimento de *slots* devido à seu desempenho superior ao modelo CRF, por exemplo, utilizando RNNs (*Recurrent Neural Networks*) [Mesnil et al. 2014] ou modelos Seq2Seq (*sequence-to-sequence*) com o mecanismo de *self-attention* [Zhao and Feng 2018].

Caso você não deseje implementar seu modelo de *slot filling* do princípio, você pode se aproveitar de bibliotecas públicas, como o spaCy[13] e o Duckling[14], que já oferecem algoritmos robustos para reconhecimento de entidades em diversos idiomas.

2.6.3. Análise de sentimentos

A Análise de Sentimentos (AS), ou Mineração de Opiniões, é o estudo computacional das opiniões, atitudes e emoções de pessoas em relação a uma entidade, por exemplo, um indivíduo, evento ou tópico [Medhat et al. 2014]. Esse é um campo em NLP, que visa identificar a polaridade do sentimento (negativo, positivo ou neutro) ou emoção (e.g., raiva, antecipação, nojo, medo, alegria, tristeza, surpresa, confiança, etc.) presente em um texto [Mohammad and Turney 2013]. Além dessas tarefas principais, a AS também é usada para detecção de toxicidade, sarcasmo, análise multilíngue de sentimentos, entre outras aplicações complexas [Zhang et al. 2018]. Vale ressaltar que neste capítulo nosso foco é em conteúdo texual, mas essa tarefa vem sendo desenvolvida para outros tipos de conteúdo, como os visuais, por exemplo, na análise de sentimentos em fotos de mídias sociais [Oliveira et al. 2020].

[13]https://spacy.io/api/entityrecognizer. Último acesso em 11 de Setembro de 2022.
[14]https://github.com/facebook/duckling. Último acesso em 11 de Setembro de 2022.

Com a crescente quantidade de dados produzidos na *web* e o aumento do poder computacional, a AS tornou-se uma das áreas de pesquisa mais ativas em NLP, permeando outras áreas além da ciência da computação, como administração e ciências sociais, uma vez que opiniões são centrais para quase todas as atividades humanas e são as principais influenciadoras do comportamento [Zhang et al. 2018]. Nesse sentido, identificar o sentimento (ou opinião) em textos é muito útil para diversas aplicações, principalmente, para aplicações com tomada de decisão em larga escala. Por exemplo, seleção de comentários agressivos em artigos de notícias para serem moderados [Jigsaw 2022], extração da opinião pública sobre um candidato ou partido político [Pang et al. 2008], priorização de respostas a avaliações negativas de produtos, para diminuir possíveis impactos negativos no comportamento de consumo do produto ou marca [Bougie et al. 2003] ou estudar o impacto do sentimento em revisões (*reviews*) de hotéis visando guiar usuários nas suas decisões [Santos et al. 2020b]. Em tarefas como essas, a AS tem duas vantagens em destaque: permite análises em larga escala e reduz a subjetividade provocada por avaliadores humanos.

A AS pode ser realizada em três níveis diferentes: (i) nível de documento; (ii) nível de frase; ou (iii) nível de aspecto [Medhat et al. 2014]. No primeiro caso, a unidade básica de onde a emoção é extraída é o documento, partindo da premissa de que todo o documento expressa a opinião sobre uma única entidade [Zhang et al. 2018]. No segundo caso, a extração é feita a nível de sentença dentro de um documento [Medhat et al. 2014], partindo da premissa de que a sentença possui uma opinião sobre uma entidade [Zhang et al. 2018]. Nos dois primeiros níveis, não é possível obter detalhes importantes para alguns tipos de aplicações em que opiniões sobre diferentes entidades coexistem em um mesmo documento ou sentença. Nesse caso, entra a AS a nível de aspectos, que visa compreender não apenas a opinião sobre a entidade como um todo, mas também sobre partes específicas da entidade, chamados de aspectos ou de alvos [Zhang et al. 2018]. Esse tipo de análise mais refinada pode ser utilizada em casos em que uma entidade pode ter opiniões diferentes para aspectos distintos. Por exemplo, considere essa avaliação de produto: "A qualidade de voz deste telefone não é boa, mas a vida útil da bateria é longa", note que ela combina um sentimento positivo sobre o aspecto "vida útil da bateria", com um sentimento negativo sobre o aspecto "qualidade de voz" [Feldman 2013, Medhat et al. 2014].

As abordagens para AS podem ser divididas em três categorias, podendo ser baseadas em (i) léxicos, (ii) aprendizado de máquina, ou (iii) híbridas [Medhat et al. 2014, Zhang et al. 2018]. Na abordagem baseada em léxicos, são construídos dicionários com termos conhecidos e relacionados a sentimentos, obtidos por meio de processos estatísticos em grandes *corpus*, possibilitando a identificação de emoções em entidades por meio da ocorrência de palavras do léxico no texto a ser analisado [Medhat et al. 2014]. Já na abordagem baseada em aprendizado de máquina, são aplicados diferentes algoritmos de aprendizado supervisionado ou não supervisionado [Medhat et al. 2014, Zhang et al. 2018]. Por último, a abordagem híbrida, que é uma combinação das duas primeiras, por exemplo, em modelos que usam aprendizado de máquina para classificação de termos em larga escala, usando como base de treinamento léxicos anotados manualmente, eliminando parte do esforço manual envolvido na criação de léxicos de grande escala [Medhat et al. 2014]. A vantagem da abordagem (i), está relacionada à in-

dependência de domínio, mas com a desvantagem de ser menos precisa e demandar maior investimento de tempo na criação do léxico [Gamon and Aue 2005]. Já a abordagem (ii), possui a vantagem de geralmente ser mais precisa, uma vez que modelos de aprendizado de máquina podem identificar regularidades no *corpus*, que a simples contagem de ocorrências de categorias emocionais no texto em um léxico não conseguiria identificar [Zhang et al. 2018].

Dentre as tarefas de AS, duas são mais comuns: a detecção de polaridade do sentimento e a detecção de emoções. No primeiro caso, são identificadas as palavras em um dado texto que refletem sentimentos positivos, negativos ou neutros. Assim, é atribuído a cada palavra "relevante", uma pontuação flutuante dentro do intervalo $[-1.0; 1.0]$, sendo -1 para negativo, 0 para neutro e 1 para positivo. Já a detecção de emoção, consiste em identificar a ocorrência ou intensidade de determinadas classes de emoção em um dado texto [Medhat et al. 2014]. Para ambos os casos, dois léxicos são comumente utilizados: (i) EmoLex (*the NRC Emotion Lexicon*) [Mohammad and Turney 2013]; e (ii) LIWC (*Linguistic Inquiry Word Count*) [Pennebaker et al. 2001], que são descritos a seguir.

2.6.3.1. EmoLex

O EmoLex é um léxico em inglês que possui associações de palavras com os sentimentos e com as oito emoções primárias da teoria de Plutchik [Plutchik 1980], sendo elas: raiva, medo, antecipação, confiança, surpresa, tristeza, alegria e desgosto [Mohammad and Turney 2013]. O EmoLex foi criado em 2013 para suprir uma lacuna da época, ainda não haviam léxicos que conseguissem mapear uma grande quantidade de termos em Inglês [Mohammad and Turney 2013]. Por isso, os autores compilaram unigramas e bigramas comuns na língua Inglesa, incluindo termos do *General Inquire* [Stone et al. 1966] e do *WordNet Affect Lexicon*, em um único léxico. Então, usaram uma plataforma *crowdsourcing* para classificação manual dos termos em larga escala, construindo um dos maiores léxicos disponíveis em língua Inglesa, feito especificamente para análise de sentimentos e emoções [Mohammad and Turney 2013].

Muito mais que o léxico, esse trabalho apresentou um arcabouço usando boas práticas para a classificação de termos em larga escala, além de considerar aspectos éticos e respectivas implicações em novos trabalhos [Hipson and Mohammad 2021], servindo como base para a criação de outros léxicos derivados, que podem ser encontrados no site de um dos autores[15].

2.6.3.2. LIWC

O LIWC é uma ferramenta amplamente aplicada para identificar características linguísticas, psicológicas e sociais em textos [Pennebaker et al. 2001]. A operação básica do LIWC se resume em percorrer cada uma das palavras em um texto, procurando sua ocorrência em um dicionário (léxico) interno, para identificar a qual categoria ela pertence e, por último, calculando o percentual de ocorrências de cada categoria no texto

[15]http://saifmohammad.com/WebPages/lexicons.html. Último acesso em 29 de Agosto de 2022.

[Pennebaker et al. 2001]. Inicialmente, o dicionário do LIWC era composto apenas pelas categorias de emoções negativas e positivas em Inglês, mas com sua evolução, foram incluídas classes gramaticais e processos psicológicos, como auto-reflexão, pensamento causal, entre outras [Tausczik and Pennebaker 2010]. Também foram criados dicionários para outras línguas, incluindo português do Brasil [Balage Filho et al. 2013][16]. Quando o LIWC foi desenvolvido, o objetivo era criar um sistema eficiente que pudesse explorar tanto os processos psicológicos quanto o conteúdo [Tausczik and Pennebaker 2010]. Com isso, duas categorias amplas de palavras foram incluídas, sendo elas as de conteúdo, que exploram características psicométricas, como emoções positivas e negativas, afeto, cognição, entre outras, e as de estilo ou função, que exploram as características gramaticais, como pronomes, preposições, artigos, conjunções, verbos auxiliares, entre outras [Tausczik and Pennebaker 2010]. Por exemplo, na frase "Foi uma noite escura e tempestuosa", as palavras "noite", "escuro" e "tempestade", são palavras de conteúdo, enquanto "isso", "era", "um" e "e" são palavras de função [Tausczik and Pennebaker 2010].

Dentre as categorias de conteúdo disponíveis no LIWC, estão as emoções positivas (e.g. amor, legal, doce, etc.) e negativas (e.g. ferido, feio, desagradável, etc.), processos sociais (filha, marido, vizinho, adulto, bebê, etc.), processos cognitivos (e.g. pensar, conhecer, causa, etc.), processos perceptivos (e.g. observar, escutar, sentir, etc.), processos biológicos (e.g. comer, sangue, dor, etc.), relatividade (e.g. chega, vai, embaixo, ontem, até, fim, etc.), preocupações sociais (e.g. auditar, igreja, cozinhar, trabalhar, mestrado, etc.), consentimento (e.g. concordar, ok, etc.), não-fluências e palavras de preenchimento (e.g. hm, er, umm, certo, etc.), entre outras [Tausczik and Pennebaker 2010]. Além dessas categorias, algumas contagens estão disponíveis, como quantidade de palavras, quantidade de palavras por sentença, percentual de palavras que foram identificadas no dicionário interno, entre outras contagens bastante úteis em diferentes aplicações [Tausczik and Pennebaker 2010].

Com esse repertório, o LIWC consegue trazer um alto detalhamento sobre o texto analisado, por isso, é um léxico amplamente usado em estudos na área da psicologia e sociologia. Inclusive, a versão em inglês do dicionário oficial do LIWC foi utilizada em projetos derivados em AS, como o *SentiStrength* [Thelwall et al. 2010], para construção de sua lista interna de palavras [Balage Filho et al. 2013]. Em relação aos dicionários em português brasileiro, existem duas versões, uma criada em 2007 (*LIWC_2007pt*) e outra, baseada na primeira, criada em 2015 (*LIWC_2015pt*), ambas avaliadas comparando com outros léxicos e entre versões, respectivamente [Balage Filho et al. 2013, Carvalho et al. 2019]. Atualmente a última versão *LIWC_2015pt* é a recomendada para trabalhos em produção [Carvalho et al. 2019].

2.6.3.3. *Perspective* API

Além de léxicos, que se baseiam na contagem de palavras em categorias específicas, métodos baseados em aprendizagem de máquina podem ser aplicados para indicar o grau em que diferentes categorias de emoções incidem em textos [Zhang et al. 2018].

[16]http://143.107.183.175:21380/portlex/index.php/pt/projetos/liwc. Último acesso em 24 de Agosto de 2022.

Uma aplicação bastante útil é a identificação de toxicidade em textos de mídias sociais, focando na incidência de categorias negativas como insulto, profanidade, ameaça, racismo, etc. [Jigsaw 2022]. Comportamentos *online* não civilizados, como assédio e abuso, desencorajam interações saudáveis, levando a conflitos e experiências desagradáveis [Kumar et al. 2018]. Um caso especial é representado por comentários grosseiros, desrespeitosos ou irracionais que podem levar os usuários a não se engajarem em discussões benéficas para o entendimento mútuo sobre um determinado assunto, esse é o conceito que define o termo "toxicidade" [Jigsaw 2022].

Dada a velocidade com que as discussões por meios de comentários crescem na *web* [Kumar et al. 2018], diferentes modelos de identificação de toxicidade em larga escala foram criados para resolver desafios específicos [Srivastava et al. 2018, Almerekhi et al. 2020]. Dentre eles, está o fato desse tipo de texto conter variados graus de sutileza inerentes à linguagem, aspectos culturais, especificidades de contexto, presença de sarcasmo e uso de figuras de linguagem, que podem mascarar a real toxicidade de um comentário. Outros desafios incluem o fato de os comentários em mídias sociais serem geralmente textos curtos, contendo erros de ortografia que ocorrem de forma esparsa no conjunto de dados [Srivastava et al. 2018]. Além disso, modelos baseados em aprendizado de máquina, tais como os usados para análise de toxicidade, tendem a ser suscetíveis a ataques adversários, nos quais um usuário mal-intencionado pode ajustar seu comentário, trocando palavras tóxicas por uma variante facilmente reconhecida por um humano, mas indetectável pelo modelo, por exemplo, substituindo a palavra "estúpido" por "st.Up1d0" [Hosseini et al. 2017]. Tal mudança na entrada pode causar um distúrbio na saída do modelo, fazendo com que um comentário tóxico seja reconhecido como não tóxico [Hosseini et al. 2017]. Um último desafio está relacionado à toxicidade ter variações consideráveis entre línguas distintas, considerando aspectos culturais e da própria forma de uso da língua [Srivastava et al. 2018].

Nesse sentido, um modelo multilíngue foi disponibilizado gratuitamente por meio de uma iniciativa da empresa Google, chamada Perspective API [Jigsaw 2022]. Esse modelo pode ser acessado por meio de uma API pública, que permite identificar diferentes categorias de toxicidade em comentários *online*. Por ser uma ferramenta amplamente adotada por alguns dos principais veículos jornalísticos internacionais para moderar comentários em seus portais, além de estar disponível de forma aberta, o Perspective API é um potencial candidato para aplicações envolvendo o estudo do comportamento abusivo em mídias sociais.

O Perspective API atribui uma pontuação contínua entre 0 e 1 para diferentes categorias de toxicidade em um comentário [Jigsaw 2022]. Uma pontuação mais alta para uma determinada categoria (ou atributo), indica uma maior probabilidade de um leitor perceber que o comentário possui este atributo [Jigsaw 2022]. Por exemplo, conforme apresentado na documentação da API, um comentário como "Você é um idiota" pode receber uma pontuação de $0,8$ para o atributo TOXICIDADE, indicando que 8 entre 10 pessoas perceberiam esse comentário como tóxico [Jigsaw 2022]. Com isso, é possível usar essa pontuação, por exemplo, para priorizar a análise manual de comentários ou até remover automaticamente comentários com pontuação acima de um determinado limite [Jigsaw 2022]. A arquitetura do Perspective API é composta por modelos multilíngues baseados em *sentence embeddings* BERT treinados em dados de fóruns *online*, que são

separados em redes neurais convolucionais (CNNs) para cada um dos idiomas suportados pelo modelo - essa separação garante que os modelos para diferentes línguas possam ser treinados separadamente de forma rápida, além de permitir maior velocidade no processamento da entrada [Jigsaw 2022]. Um dos idiomas suportados é o português, inclusive, um estudo recente mostra que utilizar textos de comentários em português do Brasil atinge melhores resultados que a sua tradução para inglês nessa API [Kobellarz and Silva 2022].

O Perspective API tem os chamados atributos de produção, testados em vários domínios e treinados em quantidades significativas de comentários anotados por humanos – esses atributos estão disponíveis para Inglês, Português e dezenas de outros idiomas [Jigsaw 2022], inclusive idiomas expressivamente mais complexos para essa tarefa, como o Mandarim, Sueco e Coreano. Dentre os atributos de produção estão [Jigsaw 2022]: TOXICITY - "Um comentário rude, desrespeitoso ou irracional que provavelmente fará com que as pessoas deixem uma discussão"; SEVERE_TOXICITY - "Um comentário que é muito odioso, agressivo, desrespeitoso, ou muito provável de fazer um usuário sair de uma discussão, ou desistir de compartilhar sua perspectiva. Este atributo é muito menos sensível a formas mais leves de toxicidade, como comentários que incluem usos positivos de palavrões"; IDENTITY_ATTACK - "Comentários negativos ou de ódio direcionados a alguém por causa de sua identidade"; INSULT - "Comentário ofensivo, inflamatório ou negativo para uma pessoa ou grupo de pessoas"; PROFANITY - "Xingamentos, palavrões ou outras linguagens obscenas, ou profanas"; THREAT - "Descreve a intenção de infligir dor, lesão ou violência contra um indivíduo, ou grupo". Além desses atributos, há outros ainda experimentais, somente em inglês, que não são recomendados para uso em produção [Jigsaw 2022].

2.7. Possíveis Aplicações

Nesta seção, são apresentadas possíveis aplicações que podem ser desenvolvidas com base no conhecimento semântico extraído a partir de dados de mídias sociais.

2.7.1. Recomendação de rotas personalizadas para Cidades Inteligentes

Muitos motoristas vêm adotando sistemas de recomendação de rotas durante seus deslocamentos diários, principalmente em grandes centros urbanos, onde a mobilidade tende a ser um desafio devido à constantes congestionamentos, interdições para obras, baixa qualidade no transporte público, entre outros problemas. Ao utilizar tais sistemas, o motorista é capaz de verificar possíveis rotas com tempos de viagem mais rápidos, levando em consideração principalmente a fluidez do trânsito nas vias que compõem as rotas sugeridas. Dois sistemas muito conhecidos são o Waze e o Google Maps, que consideram as condições de tráfego históricas e atuais, a distância a ser percorrida e, até mesmo, eventos ocasionais (como acidentes e bloqueios) reportados por seus usuários para recomendar rotas mais rápidas. No entanto, pessoas diferentes têm preferências diferentes para planejar suas rotas com base em vários aspectos, como distância percorrida, tempo de viagem, consumo de combustível, qualidade do asfalto, segurança e beleza das vias e seu entorno. Aspectos relacionados a mobilidade são normalmente estimados e apresentados aos motoristas, para que eles possam escolher a rota desejada. Enquanto outros aspectos das cidades geralmente são mais difíceis de coletar e interpretar, como o nível de segurança dos locais ou a beleza das vias, e, por isso, normalmente não são levados em consideração

para a recomendação.

Considerando o aspecto de segurança, é comum vermos notícias de motoristas que ao seguir orientações do sistema de recomendação acabaram transitando por regiões consideradas perigosas e sofreram algum tipo de incidente. Por exemplo, um casal que seguia uma rota recomendada pelo Waze foi baleado ao entrar em uma área perigosa no Rio de Janeiro, Brasil [Phillips 2017]. Episódios como esse mostram a importância de considerar o nível de segurança associado às rotas além da mobilidade.

Normalmente, o nível de segurança de uma determinada área da cidade, como um bairro ou distrito, é estimada com base nas ocorrências de crime registradas pelos órgãos oficiais, como as forças policias e secretarias de segurança pública. Para muitos municípos, é possível obter acesso a tais dados via portais da iniciativa de dados abertos. Apesar desses dados oficiais representarem uma "impressão digital" da insegurança das àreas urbanas e, por isso, serem imprescindíveis para elaboração e execução de políticas públicas, eles podem não agregar muito valor aos sistemas de recomendação de rotas por uma série de motivos, tais como:

- **Periodicidade**: a periodicidade em que os dados são disponibilizados pode não atender a necessidade do sistema de recomendação. Por exemplo, alguns municípios atualizam mensalmente os dados com novos registros de ocorrência, mas um crime que aconteceu um mês atrás pode não ter impacto em uma rota a ser sugerida no momento atual. Para o sistema de recomendação, quanto menor for o intervalo de tempo (minutos ou horas) entre a ocorrência de um crime e a obtenção dessa informação, melhor será o serviço prestado por ele, uma vez que ele pode evitar sugerir rotas no entorno da área onde o crime esteja ocorrendo, trazendo assim mais segurança aos seus usuários.
- **Isolamento de áreas**: regiões onde historicamente há mais insegurança podem ser completamente evitadas pelos sistemas de recomendação, o que potencialmente impactará tanto os motoristas, por terem que percorrer distâncias maiores para contornar tais áreas, mas também a comunidade local, que deixará de ter acesso a um fluxo maior de pessoas transitando pelo seu bairro e, assim, reduzindo sua capacidade de gerar mais renda por meio de seus serviços e produtos, e consequentemente, podendo agravar problemas econômicos e sociais dessas regiões.

Nesse sentido, melhor do que ter o conhecimento histórico sobre ocorrências de crime, para os sistemas de recomendação é mais importante detectar eventos de insegurança enquanto eles acontecem ou, pelo menos, pouco tempo depois deles terem acontecido. Desta maneira, é possível ajudar seus usuários a evitarem tais áreas enquanto elas estão inseguras. Para isso, algumas mídias sociais, como o Twitter e Facebook, podem ser uma rica fonte de dados, uma vez que seus usuários costumam postar relatos de crimes e incidentes sofridos e/ou testemunhados por eles para alertar seus contatos [Lal et al. 2020, von Nordheim et al. 2018].

Diferente dos dados oficiais, que são estruturados, manualmente gerados por especialistas e referente apenas a ocorrências de crime, o conteúdo proveniente de mídias sociais é diversificado, abrangendo diversos temas, como política, esportes, cultura, religião

e etc, tornando a tarefa de extração de conteúdo sobre um determinado tema muito desafiadora. Para endereçar esse desafio, em [Santos et al. 2020a] é apresentado um *framework* que combina diversas técnicas de NLP descritas aqui, indo desde o pré-processamento textual e geração de *word embeddings*, até detecção de comunidades em grafos, agrupamento de textos e análise de sentimentos.

Por meio do *framework*, os autores conseguiram mapear e extrair de dados de *Location-Based Social Networks* (LBSNs) algumas percepções coletivas de áreas urbanas, por exemplo, quais bairros eram considerados mais agressivos e violentos, em contraste com outros que eram considerados mais alegres, respeitosos e espetaculares. Para analisar a establidade da percepção ao longo do tempo, os autores conduziram uma análise temporal, onde foi possível identificar que a maioria das perceções se mantêm com poucas alterações independentemente do período avaliado. Também foi realizada uma análise comparativa com base em um conjunto de dados público, que contém as percepções dos voluntários sobre áreas urbanas expressas em um experimento controlado. Foi possível observar que ambos os resultados apresentam um nível de concordância muito semelhante.

Desta maneira, [Santos et al. 2020a] mostram que é possível utilizar o conhecimento extraído por esse *framework*, para realizar recomendações personalizadas considerando as preferências e necessidades dos usuários. Ou, ainda, pode-se considerar outras técnicas descritas neste trabalho além das utilizadas em [Santos et al. 2020a], para obter resultados mais relevantes para os sistemas de recomendação, uma vez que os autores não consideraram a utilização para uma aplicação em específico.

2.7.2. Polarização Política

Em situações politicamente polarizadas, a compreensão do papel dos principais atores, seu grau de influência e o tipo de conteúdo que espalham em mídias sociais é fundamental para o estudo da dinâmica deste fenômeno no ambiente *online* [Kobellarz et al. 2022]. Nesse sentido, a análise de tópicos e intenção de conteúdos disseminados por atores influentes e a respectiva reação de quem consome tais conteúdos, seja por meio de compartilhamentos ou comentários, são alguns exemplos de casos em que a análise textual pode trazer contribuições importantes para a compreensão de comportamentos polarizantes em mídias sociais.

Nesse âmbito, a hipótese dos "filtros-bolha" sugere uma intensificação da polarização gerada por mecanismos que levam indivíduos a serem expostos de forma seletiva a informações de seu interesse, com objetivo de otimizar o engajamento, em detrimento daquelas que expõe indivíduos à diversidade de opiniões [Pariser 2011]. Uma possível forma de mitigar seus efeitos, seria por meio da criação de conexões cruzadas entre grupos politicamente opostos, criando oportunidades de contato intergrupal e troca de informações [Burt 2003]. Esse é o papel dos "intermediadores", usuários capazes de criar pontes entre bolhas em uma rede social [Burt 2003, Yan 2019].

Dada a importância dos intermediadores, a pesquisa de Kobellarz *et al.* (2022) [Kobellarz et al. 2022] teve como objetivo compreender como usuários de *sites* de redes sociais se engajam com conteúdos compartilhados por intermediadores, sua fonte (domínio) e respectivos tópicos. Essa pesquisa teve como objetos de estudo a eleição presiden-

cial brasileira de 2018 e a eleição federal canadense de 2019, duas situações políticas com diferentes níveis de polarização. Os dados usados nas análises foram obtidos por meio da API de *streaming* do Twitter usando como critério de filtragem *hashtags* que contemplassem de forma equilibrada os diferentes candidatos e visões políticas. Foram coletados dados durante 6 semanas, incluindo o período antes e depois do dia de votação em cada situação eleitoral.

A polaridade de cada usuário foi inferida de acordo com a orientação política das *hashtags* aplicadas em seus *tweets* usando uma métrica chamada $P(H)$, cujo resultado é um valor no intervalo contínuo $[-1,0;+1,0]$, em que valores próximos de $-1,0$ ou $+1,0$ representam uma inclinação maior para a esquerda ou direita, respectivamente. Em seguida, foram construídas redes de retuítes para cada uma das semanas, sobre as quais foram identificados os intermediadores mais representativos por meio de uma métrica de centralidade criada pelos autores chamada "*intergroup bridging centrality*" ou, em português, "centralidade de ponte intergrupo". Essa métrica permitiu selecionar em cada semana os 100 usuários com contas verificadas que tiveram maior efetividade em distribuir conteúdos até grupos com posicionamentos políticos diferentes. Para efeitos de convenção, estes usuários foram denominados *bubble reachers*.

Em seguida, foram mantidos apenas os *retweets* feitos de conteúdos compartilhados por *bubble reachers* que contivessem *links* para *sites* de notícias, cujo conteúdo foi extraído por meio do uso de uma biblioteca de raspagem de dados chamada *news-please*[17]. O conteúdo textual das notícias foi pré-processado na seguinte ordem: (1) remoção de *stop-words* de listas específicas para os idiomas de cada conjunto de dados, sendo português para o caso brasileiro e inglês para o caso canadense, usando a biblioteca NLTK[18]; (2) remoção de *tokens* que não fossem substantivos, nomes próprios, adjetivos ou verbos usando modelos de *POS-tagging* pré-treinados específicos para os idiomas de cada conjunto de dados usando a biblioteca Spacy[19]; (3) *stemming* de *tokens* utilizando modelos pré-treinados para os idiomas de cada país através da biblioteca Spacy; (4) transformação de *tokens* em minúsculas e remoção de caracteres especiais e acentos; (5) extração de bi e trigramas usando a biblioteca Gensim[20].

Após a limpeza dos dados, os tópicos foram extraídos por meio da aplicação do algoritmo Latent Dirichlet Allocation (LDA) [Blei et al. 2003] usando a implementação da biblioteca Gensim. Este algoritmo permite identificar tópicos em um conjunto de textos, considerando que cada texto é composto por uma mistura de tópicos [Blei et al. 2003]. Como a escolha de tópicos é feita manualmente para o LDA, foram criados múltiplos modelos para cada conjunto de dados com o número de tópicos indo de 1 até 50. Dentro desse intervalo, foi escolhido o modelo cuja quantidade de tópicos teve o maior grau de similaridade semântica entre os textos por meio da métrica de coerência gerada com o modelo C_v[22], que é baseado na similaridade indireta do cosseno das palavras mais representativas de cada tópico em todos os documentos do conjunto de dados. Para todos os casos, foi utilizada a mesma semente para que os resultados da identificação dos tópicos pudessem ser replicados. Com esse método, foram identificados 48 tópicos no Brasil e 26

[17]https://pypi.org/project/news-please. Último acesso em 11 de Setembro de 2022.
[18]https://www.nltk.org. Último acesso em 11 de Setembro de 2022.
[19]https://spacy.io. Último acesso em 11 de Setembro de 2022.
[20]https://radimrehurek.com/gensim. Último acesso em 11 de Setembro de 2022.

no Canadá dentre as 338 e 484 notícias obtidas em cada uma desses países. Após a extração de tópicos com este método, foi analisada a dominância de tópicos de cada conteúdo, a partir do qual foi verificado que 87% e 68% dos conteúdos nos conjuntos de dados brasileiro e canadense, respectivamente, eram dominados por apenas um tópico, com um domínio de pelo menos 80% sobre outros tópicos. Considerando esse resultado, foi extraído apenas o tema dominante de cada conteúdo, verificando-se a quantidade média de conteúdos ligados a cada tema dominante. No caso brasileiro, encontrou-se uma quantidade média de $6,5$ ($\sigma = 2,5$) conteúdos por tema dominante e, no caso canadense, uma média de $15,0$ ($\sigma = 4,1$) conteúdos por tema dominante. Esses resultados indicam que a maioria dos tópicos estava bem definida (menos nebulosa) e distribuída uniformemente entre os conteúdos. Por fim, os autores avaliaram manualmente todos os tópicos identificados e chegaram a um consenso sobre se os tópicos estavam diretamente relacionados às respectivas situações políticas.

Para seguir com as análises, foram definidas três entidades de interesse: (i) o conteúdo, representado pelo *link* da notícia, (ii) o domínio, representando a fonte que publicou o conteúdo e (iii) o tópico, representado pelo tópico latente dominante ligado ao conteúdo. Assim como os usuários, as entidades também tiveram sua polaridade estimada. Para isso foi calculada a polaridade ($P(H)$) média ponderada dos usuários que compartilharam a respectiva entidade, gerando uma métrica chamada $RP(H)$ (polaridade relativa), cuja interpretação é a mesma da métrica $P(H)$. Dentre os resultados foi identificado que dentre os *bubble reachers* mais representativos, estavam contas de veículos jornalísticos, jornalistas, personalidades políticas e ativistas políticos, mas somente os veículos jornalísticos neutros conseguiram sustentar essa posição durante mais de uma semana. Isso indica que esses usuários conseguiram alcançar grupos homófilos distintos com maior eficiência na rede, ao passo que sustentaram essa posição ao longo das semanas no período eleitoral. Quanto ao comportamento de usuários no compartilhamento de notícias apontadas por *links* em tweets criados por *bubble reachers*, usando testes de correlação, foi identificado que domínios com RP(H) neutro representando veículos jornalísticos conseguiram distribuir seus conteúdos de forma balanceada para usuários em grupos politicamente opostos. Apesar disso, mesmo expostos a conteúdos politicamente diversificados, usuários polarizados preferem se engajar mais frequentemente com conteúdos que favorecem seu posicionamento político. Em relação aos tópicos, não foi identificada correlação entre a polaridade de tópicos e o posicionamento político dos usuários, sendo importante apenas a polaridade do conteúdo nesse sentido.

Os resultados indicaram que, mesmo contornando os mecanismos de filtro-bolha, a criação de conexões cruzadas entre grupos distintos pode não ser suficiente para reduzir a polarização de uma rede. A evidência de que veículos de notícias tendem a distribuir com mais eficiência conteúdos a grupos polarizados pode ser interpretada no sentido de que a mídia jornalística tenta manter seu propósito de imparcialidade, pelo menos quando considerado o "viés de seleção" (*gatekeeping bias*) [D'Alessio and Allen 2000]. Esse é um ponto importante no momento atual, em que a mídia jornalística sofre constantes ataques por líderes políticos e seus seguidores, sob a alegação de que produz notícias falsas ou tendenciosas para enfraquecer seu poder [21].

[21] https://fenaj.org.br/violencia-contra-jornalistas-cresce-10577-em-2020-com-jair-bolsonaro-liderando-ataques e https://cpj.org/wp-content/uploads/2020/04/cpj. Último acesso em 11 de Setembro de 2022.

2.8. Conclusão

As mídias sociais são valiosas fontes de informação, devido ao expressivo volume de dados gerados por seus usuários e a facilidade em acessá-los de maneira programática em larga escala. E, como boa parte do conteúdo dessas fontes são textos escritos em linguagem natural, torna-se fundamental a utilização de técnicas de processamento de linguagem natural em conjunto com modelos de aprendizado de máquina para processar, analisar e extrair *insights* relevantes a partir de tais textos, que podem auxiliar no desenvolvimento de novas aplicações e serviços. Diante disso, este capítulo além de apresentar as características de famosas mídias sociais e como coletar seus dados, também introduziu os fundamentos e ferramentas das principais etapas do processo de análise de textos, fornecendo, assim, os meios necessários para desenvolver aplicações que possam tirar proveito do conhecimento semântico e emocional extraídos a partir de dados de mídias sociais. Apesar do foco em textos de mídias sociais e suas particularidades, boa parte das técnicas apresentadas também podem ser aplicadas em outras fontes textuais.

2.9. Reconhecimentos

Este capítulo foi parcialmente financiado pela CAPES - Finance Code 001, projeto GoodWeb (Bolsa 2018/23011-1 da Fundação de Amparo à Pesquisa de São Paulo - FAPESP), e CNPq (bolsa 310998/2020-4).

Referências

[Aggarwal and Zhai 2012] Aggarwal, C. C. and Zhai, C. (2012). A survey of text clustering algorithms. In *Mining text data*, pages 77–128. Springer.

[Aiello et al. 2013] Aiello, L. M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Göker, A., Kompatsiaris, I., and Jaimes, A. (2013). Sensing trending topics in twitter. *IEEE Transactions on multimedia*, 15(6):1268–1282.

[Alghamdi and Alfalqi 2015] Alghamdi, R. and Alfalqi, K. (2015). A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, 6(1).

[Almerekhi et al. 2020] Almerekhi, H., Kwak, H., Salminen, J., and Jansen, B. J. (2020). *Are These Comments Triggering? Predicting Triggers of Toxicity in Online Discussions*, page 3033–3040. Association for Computing Machinery, New York, NY, USA.

[Artetxe and Schwenk 2019] Artetxe, M. and Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

[Balage Filho et al. 2013] Balage Filho, P., Pardo, T. A. S., and Aluísio, S. (2013). An evaluation of the brazilian portuguese liwc dictionary for sentiment analysis. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*.

[Barbier and Liu 2011] Barbier, G. and Liu, H. (2011). Data mining in social media. In *Social network data analytics*, pages 327–352. Springer.

[Bird 2006] Bird, S. (2006). Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics.

[Blei et al. 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

[Bojanowski et al. 2017] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

[Bougie et al. 2003] Bougie, R., Pieters, R., and Zeelenberg, M. (2003). Angry customers don't come back, they get back: The experience and behavioral implications of anger and dissatisfaction in services. *J. of the acad. of mark. science*, 31(4):377–393.

[Bowman et al. 2015] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

[Burt 2003] Burt, R. S. (2003). The social structure of competition. *Networks in the knowledge economy*, 13:57–91.

[Carvalho et al. 2019] Carvalho, F., Rodrigues, R. G., Santos, G., Cruz, P., Ferrari, L., and Guedes, G. P. (2019). Evaluating the brazilian portuguese version of the 2015 liwc lexicon with sentiment analysis in social networks. In *Anais do BRASNAM*.

[Cer et al. 2017] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

[Cer et al. 2018] Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

[Chidambaram et al. 2018] Chidambaram, M., Yang, Y., Cer, D., Yuan, S., Sung, Y.-H., Strope, B., and Kurzweil, R. (2018). Learning cross-lingual sentence representations via a multi-task dual-encoder model. *arXiv preprint arXiv:1810.12836*.

[Churchill and Singh 2021] Churchill, R. and Singh, L. (2021). The evolution of topic modeling. *ACM Comput. Surv.*

[Cody et al. 2015] Cody, E. M., Reagan, A. J., Mitchell, L., Dodds, P. S., and Danforth, C. M. (2015). Climate change sentiment on twitter: An unsolicited public opinion poll. *PloS one*, 10(8):e0136092.

[Conneau et al. 2017] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proc of EMNLP*, pages 670–680, Copenhagen, Denmark.

[Conover et al. 2011] Conover, M., Ratkiewicz, J., Francisco, M., Gonçalves, B., Menczer, F., and Flammini, A. (2011). Political polarization on twitter. In *Proceedings of the international aaai conference on web and social media*, volume 5, pages 89–96.

[D'Alessio and Allen 2000] D'Alessio, D. and Allen, M. (2000). Media bias in presidential elections: A meta-analysis. *Journal of communication*, 50(4):133–156.

[Devlin et al. 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Dib 2022] Dib, F. (2022). Regular expressions 101.

[Feldman 2013] Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.

[Feng et al. 2020] Feng, F., Yang, Y., Cer, D., Arivazhagan, N., and Wang, W. (2020). Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

[Fleiss et al. 1981] Fleiss, J. L., Levin, B., Paik, M. C., et al. (1981). The measurement of interrater agreement. *Stat meth rat and prop*, 2(212-236).

[Gamon and Aue 2005] Gamon, M. and Aue, A. (2005). Automatic identification of sentiment vocabulary: Exploiting low association with known sentiment terms. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing*, pages 57–64, Ann Arbor, Michigan. Association for Computational Linguistics.

[González-Bailón and De Domenico 2021] González-Bailón, S. and De Domenico, M. (2021). Bots are less central than verified accounts during contentious political events. *Proceedings of the National Academy of Sciences*, 118(11).

[Harris 1954] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

[Hipson and Mohammad 2021] Hipson, W. E. and Mohammad, S. M. (2021). Emotion dynamics in movie dialogues. *PloS one*, 16(9):e0256153.

[Hofmann 2001] Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1):177–196.

[Honnibal et al. 2020] Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.

[Hosseini et al. 2017] Hosseini, H., Kannan, S., Zhang, B., and Poovendran, R. (2017). Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*.

[Huh and Fienberg 2012] Huh, S. and Fienberg, S. E. (2012). Discriminative topic modeling based on manifold learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):1–25.

[Indurkhya and Damerau 2010] Indurkhya, N. and Damerau, F. J. (2010). *Handbook of natural language processing*. Chapman and Hall/CRC.

[Iyyer et al. 2015] Iyyer, M., Manjunatha, V., Boyd-Graber, J., and Daumé III, H. (2015). Deep unordered composition rivals syntactic methods for text classification. In *Proc. of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing*, pages 1681–1691.

[Jigsaw 2022] Jigsaw, G. (2022). Perspective api. Acessed May 31, 2022.

[Joulin et al. 2016] Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

[Jurafsky and Martin 2021] Jurafsky, D. and Martin, J. H. (2021). *Speech and language processing 3. Ed.*, volume 3. Pearson London.

[Kemp 2022] Kemp, S. (2022). Digital 2022: July global statshot report. https://datareportal.com/reports/digital-2022-july-global-statshot. Accessed: 2022-09-08.

[Kiros et al. 2015] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. *Advances in neural information processing systems*, 28.

[Kleene et al. 1956] Kleene, S. C. et al. (1956). Representation of events in nerve nets and finite automata. *Automata studies*, 34:3–41.

[Kobellarz and Silva 2022] Kobellarz, J. and Silva, T. H. (2022). Should we translate? evaluating toxicity in online comments when translating from portuguese to english. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, Curitiba, Brasil.

[Kobellarz et al. 2022] Kobellarz, J. K., Brocic, M., Graeml, A. R., Silver, D., and Silva, T. H. (2022). Reaching the bubble may not be enough: news media role in online political polarization. *arXiv*.

[Kumar et al. 2018] Kumar, S., Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2018). Community interaction and conflict on the web. In *Proc of WWW*, WWW '18, page 933–943, Republic and Canton of Geneva, CHE.

[Ladeira et al. 2022] Ladeira, L. Z., Santos, F., Cléopas, L., Buteneers, P., and Villas, L. (2022). Neo-nda: Neo natural language data augmentation. In *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, pages 99–102. IEEE.

[Lafferty et al. 2001] Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ACM International Conference on Machine Learning*, pages 282–289.

[Lal et al. 2020] Lal, S., Tiwari, L., Ranjan, R., Verma, A., Sardana, N., and Mourya, R. (2020). Analysis and classification of crime tweets. *Procedia Computer Science*, 167:1911–1919. International Conference on Computational Intelligence and Data Science.

[Landauer et al. 1998] Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

[MacQueen 1967] MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297.

[Medhat et al. 2014] Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

[Mesnil et al. 2014] Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D., et al. (2014). Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

[Messias et al. 2013] Messias, J., Schmidt, L., Oliveira, R., and Benevenuto, F. (2013). You followed my bot! transforming robots into influential users in twitter. *First Monday*.

[Mikolov et al. 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mohammad and Turney 2013] Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

[Murtagh and Contreras 2012] Murtagh, F. and Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97.

[Oliveira et al. 2020] Oliveira, W. B. d., Dorini, L. B., Minetto, R., and Silva, T. H. (2020). Outdoorsent: Sentiment analysis of urban outdoor images by using semantic and deep features. *ACM Trans. Inf. Syst.*, 38(3).

[Omran et al. 2007] Omran, M. G., Engelbrecht, A. P., and Salman, A. (2007). An overview of clustering methods. *Intelligent Data Analysis*, 11(6):583–605.

[Palla et al. 2005] Palla, G., Derényi, I., Farkas, I., and Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *nature*, 435(7043):814.

[Pang et al. 2008] Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135.

[Pariser 2011] Pariser, E. (2011). *The Filter Bubble: What The Internet Is Hiding From You*. Penguin Books Limited.

[Pennebaker et al. 2001] Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

[Pennington et al. 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*.

[Phillips 2017] Phillips, D. (2017). How directions on the waze app led to death in brazil's favelas. https://goo.gl/QxHdKv. Accessed: 2022-08-29.

[Plutchik 1980] Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier.

[Pombo 2022] Pombo, O. (2022). Morte de sócrates.

[Porter 1980] Porter, M. F. (1980). An algorithm for suffix stripping. *Program*.

[Proferes et al. 2021] Proferes, N., Jones, N., Gilbert, S., Fiesler, C., and Zimmer, M. (2021). Studying reddit: A systematic overview of disciplines, approaches, methods, and ethics. *Social Media+ Society*, 7(2):20563051211019004.

[Radford et al. 2018] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training. *OpenAI Blog*.

[Ramage et al. 2009] Ramage, D., Rosen, E., Chuang, J., Manning, C. D., and McFarland, D. A. (2009). Topic modeling for the social sciences. In *NIPS 2009 workshop on applications for topic models: text and beyond*, volume 5, pages 1–4.

[Řehůřek and Sojka 2010] Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proc. of LREC, Workshop*, pages 45–50, Valletta, Malta. ELRA.

[Reimers and Gurevych 2019] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

[Reimers and Gurevych 2020] Reimers, N. and Gurevych, I. (2020). Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

[Rendel et al. 2016] Rendel, A., Fernandez, R., Hoory, R., and Ramabhadran, B. (2016). Using continuous lexical embeddings to improve symbolic-prosody prediction in a text-to-speech front-end. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE.

[Robertson 2004] Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*.

[Röder et al. 2015] Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408.

[Rodrigues 1997] Rodrigues, N. (1997). *Flor de obsessão: as 1000 melhores frases de Nelson Rodrigues*, volume 12. Companhia

das Letras.

[Rogers 2009] Rogers, R. (2009). *The end of the virtual: Digital methods*, volume 339. Amsterdam University Press.

[Santos et al. 2020a] Santos, F. A., Silva, T. H., Loureiro, A. A., and Villas, L. A. (2020a). Automatic extraction of urban outdoor perception from geolocated free texts. *Social Network Analysis and Mining*, 10(1):1–23.

[Santos et al. 2020b] Santos, G., Mota, V. F. S., Benevenuto, F., and Silva, T. H. (2020b). Neutrality may matter: sentiment analysis in reviews of Airbnb, Booking, and Couchsurfing in Brazil and USA. *Social Network Analysis and Mining*, 10(1):45.

[Schütze et al. 2008] Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

[Sculley 2010] Sculley, D. (2010). Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178.

[Silva et al. 2019] Silva, T. H., Viana, A. C., Benevenuto, F., Villas, L., Salles, J., Loureiro, A., and Quercia, D. (2019). Urban computing leveraging location-based social network data: A survey. *ACM Comput. Surv.*, 52(1):17:1–17:39.

[Sivakumar et al. 2020] Sivakumar, S., Videla, L. S., Kumar, T. R., Nagaraj, J., Itnal, S., and Haritha, D. (2020). Review on word2vec word embedding neural net. In *Proc of ICOSEC*, pages 282–290. IEEE.

[Socher et al. 2013] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc of EMNLP*, pages 1631–1642.

[Sparck Jones 1972] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

[Srivastava et al. 2018] Srivastava, S., Khurana, P., and Tewari, V. (2018). Identifying aggression and toxicity in comments using capsule network. In *Proc of TRAC*, pages 98–105, Santa Fe, New Mexico, USA.

[Stats 2022] Stats, I. L. (2022). Twitter usage statistics. https://www.internetlivestats.com/twitter-statistics/. Accessed: 2022-09-08.

[Stone et al. 1966] Stone, P. J., Dunphy, D. C., and Smith, M. S. (1966). The general inquirer: A computer approach to content analysis. *Journal of Regional Science*.

[Tan et al. 2018] Tan, P.-N., Steinbach, M., and Kumar, V. (2018). *Introduction to data mining*. Pearson Education, 2nd edition.

[Tausczik and Pennebaker 2010] Tausczik, Y. R. and Pennebaker, J. W. (2010). The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54.

[Terragni et al. 2021] Terragni, S., Fersini, E., Galuzzi, B. G., Tropeano, P., and Candelieri, A. (2021). OCTIS: Comparing and optimizing topic models is simple! In *Proc. of the 16th Conference of the European Chapter of the ACL*, pages 263–270.

[Thelwall et al. 2010] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American society for information science and technology*, 61(12):2544–2558.

[Thompson 1968] Thompson, K. (1968). Programming techniques: Regular expression search algorithm. *Communications of the ACM*, 11(6):419–422.

[Tufekci 2014] Tufekci, Z. (2014). Big questions for social media big data: Representativeness, validity and other methodological pitfalls. In *Eighth international AAAI conference on weblogs and social media*.

[Ushioda 1996] Ushioda, A. (1996). Hierarchical clustering of words and application to nlp tasks. In *Fourth Workshop on Very Large Corpora*.

[Vaswani et al. 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Vayansky and Kumar 2020] Vayansky, I. and Kumar, S. A. (2020). A review of topic modeling methods. *Information Systems*, 94:101582.

[von Nordheim et al. 2018] von Nordheim, G., Boczek, K., and Koppers, L. (2018). Sourcing the sources. *Digital Journalism*, 6(7):807–828.

[Wang and McCallum 2006] Wang, X. and McCallum, A. (2006). Topics over time: a non-markov continuous-time model of topical trends. In *Proc. of the 12th ACM SIGKDD*, pages 424–433.

[Weedon et al. 2017] Weedon, J., Nuland, W., and Stamos, A. (2017). Information operations and facebook. *Retrieved from Facebook: https://fbnewsroomus. files. wordpress. com/2017/04/facebook-and-information-operations-v1. pdf*.

[Wen et al. 2016] Wen, Z., Li, Y., and Tao, J. (2016). The parameterized phoneme identity feature as a continuous real-valued vector for neural network based speech synthesis. In *INTERSPEECH*.

[Yan 2019] Yan, P. (2019). Information bridges: Understanding the informational role of network brokerages in polarised online discourses. In *Proc. of ICIS*, pages 377–388. Springer.

[Yang et al. 2017] Yang, X., Chen, Y.-N., Hakkani-Tür, D., Crook, P., Li, X., Gao, J., and Deng, L. (2017). End-to-end joint learning of natural language understanding and dialogue manager. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5690–5694. IEEE.

[Yang et al. 2019] Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Abrego, G. H., Yuan, S., Tar, C., Sung, Y.-H., et al. (2019). Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*.

[Zhang et al. 2021] Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.

[Zhang et al. 2018] Zhang, L., Wang, S., and Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

[Zhao and Feng 2018] Zhao, L. and Feng, Z. (2018). Improving slot filling in spoken language understanding with joint pointer and attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 426–431.

[Zhao and Karypis 2002] Zhao, Y. and Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524.

[Zhu and Ghahramani 2002] Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. *Technical Report*.

[Zimmer 2020] Zimmer, M. (2020). "but the data is already public": on the ethics of research in facebook. In *The Ethics of Information Technologies*, pages 229–241. Routledge.

[Zipf 2016] Zipf, G. K. (2016). *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books.

Capítulo 3

Polarização em Redes Sociais: Conceitos, Aplicações e Desafios

Bruno Hott[1], Bruno P. Santos[1,2], Túlio Corrêa Loures[1], Fabrício Benevenuto[1] e Pedro O.S. Vaz-de-Melo[1]

[1]Departamento de Ciência da Computação, UFMG

[2]Departamento de Ciência da Computação, UFBA

{brunohott,loures.tc,fabricio,olmo}@dcc.ufmg.br, bruno.ps@ufba.br

Abstract

The polarization assessed in social networks has reflected the predisposition of society to the clash of ideas and the recent encouragement of political rivalry in the world. From this context, several questions are raised, such as: Are people becoming more polarized? If so, what are the positive and negative impacts of social media on this process? Is it possible to measure polarization in social networks? The goal of this tutorial is to discuss the current scenario of research in polarization, through a critical overview of the area, its challenges and opportunities. For such, the main concepts and definitions of polarization will be presented. The flow of data collection on polarization, its processing, analysis and knowledge extraction will also be presented. For the latter, a special focus will be given to a taxonomy proposal for polarization metrics in social networks. At end, we will exercise our knowledge in a practical analysis of polarization applied to the Covid-19 topic.

Resumo

A polarização aferida em redes sociais tem refletido a predisposição da sociedade para o embate de ideias e o recente incentivo a rivalidade política no mundo. Deste contexto, diversas questões são levantadas, tais como: As pessoas estão se tornando mais polarizadas? Em caso afirmativo, quais são os impactos positivos e negativos das redes sociais neste processo? É possível medir polarização nas redes sociais? Neste minicurso, o objetivo é discutir o atual cenário de pesquisa em polarização, através de uma visão crítica geral da área, seus desafios e oportunidades. Para tal, os principais conceitos e definições sobre polarização serão apresentados. Assim como o fluxo de coleta dados sobre polarização, seu processamento, análises e extração de conhecimento. Para este último, será dado enfoque especial em uma proposta de taxonomia para métricas de polarização em redes sociais. Ao final, exercitaremos nossos conhecimentos em uma análise prática de polarização aplicada ao tópico Covid-19.

3.1. Introdução

Incipit Vita Nova (uma vida nova começa) para estudos de polarização com o advento da Internet. Essa ferramenta transformou o modo como interagimos e nos comunicamos e, especialmente, potencializou a disseminação de opiniões individuais e coletivas. A polarização tem ganhado enfoque especial da academia ao longo do tempo e, mais recentemente [Moreira et al., 2020; Valensise et al., 2022], também da indústria [Kubin and von Sikorski, 2021], devido ao seu impacto (positivo e negativo) potencial na sociedade (por exemplo, através da economia e política). Este capítulo aborda a Polarização por meio de uma perspectiva teórica e prática. O conteúdo aqui apresentado explora a estrutura, organização, desafios e aplicações dos estudos de polarização com enfoque na Internet, especialmente nas mídias sociais. Nesta seção, iniciamos discussões conceituais sobre polarização, motivamos o leitor com casos recentes e históricos, bem como apresentamos uma proposta de metodologia que guia o capítulo. Para iniciar a discussão, será levantada a seguinte questão: *o que é Polarização?*

A gêneses da polarização moderna aparece ao expressarmos opiniões, desejos ou intenções em veículos propagadores que variam desde a TV até *web blogs* pessoais. No Brasil, por exemplo, os candidatos(as) à presidência do país afirmam e levantam evidências, em debate na TV[1], que as eleições presidenciais de 2022 estão polarizadas. Na Internet, o Facebook tem sido acusado de promover conteúdos que dividem a população, podendo criar ou acelerar polarizações políticas[2]. No caso das mídias sociais, uma motivação da promoção destes conteúdos seria explorar a pré-disposição do cérebro humano em consumir conteúdos que reforçam sua visão de mundo [Klayman, 1995]. Desta forma, a hipótese é que as plataformas estão entregando cada vez mais conteúdos extremos com o objetivo de que os usuários fiquem por mais tempo na plataforma.

Alguns tópicos são naturalmente alvos de debate, como esportes, política, consumo de drogas, entre outros. Empiricamente, essas discussões tendem a fazer com que as pessoas movam suas opiniões para versões mais extremas delas mesmas [Sunstein, 1999]. Por exemplo, aqueles com opiniões contrárias à regulamentação das drogas se tornarão extremamente contrárias depois de interagir com quem compartilha sua visão. As pessoas cada vez mais estão ouvindo suas próprias vozes sendo ecoadas por seus semelhantes. Como consequência, se torna mais difícil para que a população possa resolver os problemas que a sociedade enfrenta conjuntamente, como o aquecimento global, por exemplo [Sunstein, 2018].

Por um lado, a Internet e, em especial, as mídias sociais vem sendo apontadas como estimuladoras da polarização entre os americanos nas últimas décadas ao criarem as chamadas "câmaras de eco" [Iyengar et al., 2012; Lelkes, 2016]. Nessas câmaras, as pessoas são estimuladas por pensamentos análogos, o que as isolam de divergências de opiniões e pensamentos [Bright, 2017; Lima et al., 2018]. Mas é preciso cautela, pois outros autores afirmam que o aumento da polarização pode não estar diretamente ligada ao uso das mídias sociais. Por exemplo, segundo [Boxell et al., 2017], os níveis de

[1]Primeiro debate presidencial 2022 organizado por um pool de emissoras formado entre o Grupo Bandeirantes de Comunicação, a TV Cultura, o jornal Folha de S. Paulo e o portal Uol.

[2]`https://www.washingtonpost.com/opinions/2020/10/26/facebook-algorithm-conservative-liberal-extremes/`

polarização crescem mais em populações que tipicamente não usam medias sociais como, por exemplo, a população idosa.

Segundo o dicionário Priberam[3], polarização é a concentração de ideias em um polo que se opõe a outro. Também existe uma vertente nas ciências sociais que indica que a polarização se dá quando membros de grupos da sociedade se movem na direção dos extremos [Fiorina et al., 2008]. Há quem defina polarização como um processo social, onde grupos se dividem em dois sub-grupos opostos cada qual com posições conflitantes uma as outras, com alguns poucos indivíduos neutros [Sunstein, 1999].

As definições possuem em comum a ideia de que a polarização divide um grupo social por meio da divergência de ideias. E é com esse conceito em mente que trabalharemos ao longo do capítulo. Vale notar que o enfoque deste trabalho se dá no recorte técnico de sua identificação. A análise completa dos impactos sociais e desdobramentos advindos da polarização vão muito além do escopo deste trabalho.

3.1.1. Perspectiva Histórica

Polarização não é um tópico de pesquisa novo, embora tenha recebido grande atenção acadêmica e industrial nos últimos anos. A Figura 3.1 exibe uma linha do tempo de artigos publicados na base de dados *Web of Science* contendo a conjunção das seguintes palavras-chave: *'Political'*, *'Polarization'* e *'Media'*. Estas chaves estão diretamente relacionadas ao conteúdo aqui abordado, apesar de filtrarem somente um recorte dos trabalhos referentes ao tópico polarização.

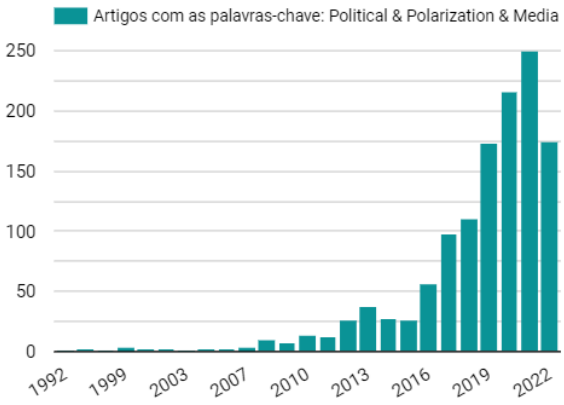


Figura 3.1: Artigos no Web of Science[4]

O artigo mais antigo presente em nossa busca a influência de variáveis religiosas em questões associadas ao aborto aborto [Woodrum and Davison, 1992], um tema ainda controverso em 2022 [Roy and Goldwasser, 2020]. Em contrapartida, o artigo mais recente é de julho de 2022, em que Borah and Singh [2022] realizam investigações sobre como a rede social Twitter tem sido usada para formar enlaces de comunicação inter e extra partidos e seus desdobramentos na divergência de opiniões no discuso político.

O estudo da polarização atrai pesquisadores por diversos motivos. Primeiro pelo valor agregado como, por exemplo, o impacto social na identificação de polarização, no combate a desinformação e informação falsa, na proposta de técnicas para redução da polarização, entre outros. Segundo, por quê a temática é inerentemente multidisciplinar, o que cabe percepções e abordagens diferentes sobre a problemática. Terceiro, por quê há um interseção importante entre a academia e a indústria sobre a temática que busca por soluções cientes da polarização – vide o recente caso da recomendação de conteúdos do

[3] `https://dicionario.priberam.org`

[4] Dados coletados até Agosto de 2022.

Facebook[5].

Do ponto de vista da computação, há diversas frentes de estudo que residem na interseção entre disciplinas distintas. Por exemplo, a aplicação de técnicas de inteligência computacional na identificação, previsão e controle da polarização [Belcastro et al., 2020; Garimella et al., 2021; Ribeiro et al., 2019; Tokita et al., 2021]. Uma lista não exaustiva de áreas de pesquisa que trabalham com a caracterização da polarização dentro do contexto da computação são: i) coleta e processamento de bases de dados [Garimella et al., 2018b]; ii) mineração de opiniões e análise de sentimentos [Rathje et al., 2021], entre outros, com o objetivo de classificar o viés dos usuários ou conteúdo [Bakshy et al., 2015; Garimella et al., 2021; Weld et al., 2021]; iii) desenvolvimento de modelos estatísticos, da teoria dos grafos, da inteligência artificial, entre outros, com o objetivo de classificar e quantificar a polarização [Garimella et al., 2018b; Pergola et al., 2020; Vicario et al., 2019]; iv) visualização e análise de resultados [Jang and Allan, 2018; Roy and Goldwasser, 2020].

3.1.2. Uma taxonomia para o estudo da polarização

Indicamos que a polarização pode ser caracterizada pela divisão das entidades de um grupo em duas partições com posições conflitantes. Logo, antes de analisarmos a polarização de uma população, precisamos definir a posição ou o *viés* de cada uma das entidades que a compõe.

Viés é definido como posicionamento, ou apoio, de um usuário ou declaração com relação a um tópico específico. Por exemplo, ao escolhermos o tópico "legalização das drogas", teremos um conjunto de indivíduos com viés pró-legalização e outro com viés contrário à legalização. Formalizamos o cálculo do viés como mostrado na Equação 1, onde X representa a declaração ou indivíduo a ser analisado com relação a um tópico específico (T). Como saída temos um dos três rótulos: $\{negativo, positivo, neutro\}$.

$$V(X \mid T) = \{\text{Negativo}, \text{Positivo}, \text{Neutro}\} \tag{1}$$

É importante ressaltar que a representação do viés pode ser feita de diferentes formas, como posicionamento binário, discreto de múltiplos níveis (como na Equação 1), ou numérico contínuo, por exemplo, um número real no intervalo entre $[-1;1]$.

O viés pode ser calculado para um indivíduo (ex.: usuário de uma rede social) ou ainda para um conteúdo. Por exemplo, poderemos encontrar um texto em uma página de notícias com viés claramente favorável ao tema de liberação de drogas, em contraponto a uma notícia de outro jornal que aborda o tema com viés negativo. Neste capítulo abordaremos somente viés baseado em conteúdos textuais, porém o mesmo se aplica para outros tipos de mídia, como imagens ou vídeos, por exemplo.

A Figura 3.2 apresenta uma proposta de taxonomia para o estudo sobre polarização e viés. O viés é calculado individualmente para cada usuário ou conteúdo da rede social. Por sua vez a polarização é uma métrica de grupos, que pode ser um grupo de usuários ou um grupo de conteúdos em torno de um tema. Dessa forma fica claro que as métricas ou ferramentas utilizadas para cálculo do viés serão diferentes daquelas para cálculo da polarização. A Figura 3.2 ainda mostra, como exemplo, que o viés pode ser extraído por meio de questionários (*surveys*) passados para cada usuário ou por métodos

[5]Observar nota de rodapé 2

computacionais, como por meio da extração de *hashtags* com vieses conhecidos contidos em*tweets*. As métricas de polarização podem ser calculadas utilizando métricas estatísticas [Akhtar et al., 2019; Morales et al., 2015], de teoria dos grafos [Garimella et al., 2018b] ou ainda por meio de técnicas de inteigência computacional [Al Amin et al., 2017; Roy and Goldwasser, 2020], tais como aprendizado de máquina, ciência dos dados, mineração de informação, entre outras.

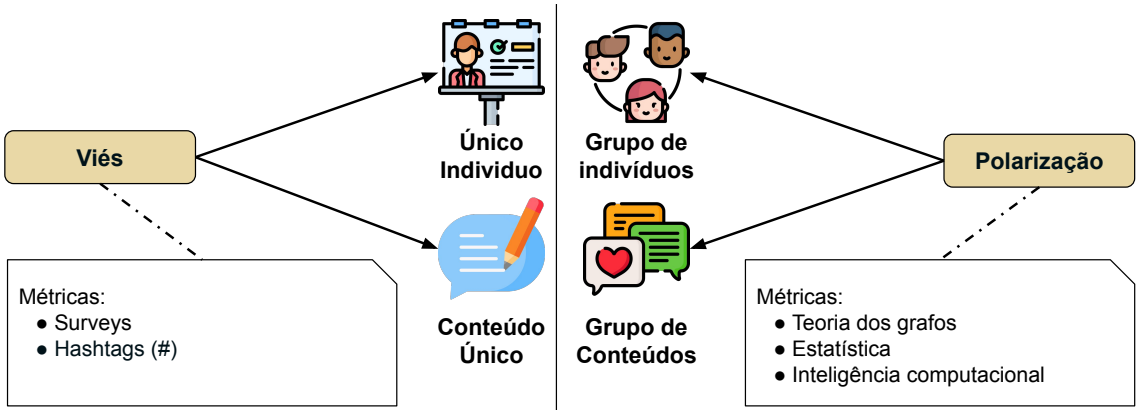


Figura 3.2: Taxonomia proposta de viés e polarização

3.1.3. Metodologia: um mapa do Capítulo

Apresentamos uma metodologia de quatro estágios, como mostrado na Figura 3.3, para realizar análises de polarização em mídias sociais, a saber: i) *Dados sobre Polarização*; ii) *Viés de entidades dos dados*; iii) *Polarização de grupo* e; iv) *Análise de polarização*. A seguir, apresentaremos uma breve descrição de cada estágio e, nas seguintes seções do capítulo aprofundamos discussões sobre cada etapa.

Figura 3.3: Visão geral do Capítulo

Dados sobre Polarização. O propósito deste estágio é capturar e pré-processar os dados referentes a polarização. Dados sobre polarização tipicamente são valores discretos, os quais são derivados de pessoas que expressam seus pensamentos, desejos ou opiniões através de meios de comunicação e interações. No contexto de polarização em mídias sociais, por exemplo, pode-se capturar e modelar dados correspondentes a um tópico de conversação juntamente com um conjunto de entidades relacionadas a ele. Um tópico pode representar um tema ou um assunto de discussão e pode ser operacionalizado (isto é, capturar dados brutos) através de um conjunto de palavras-chave ou *hashtags*, uma comunidade, ou um conteúdo, entre outros. As entidades relacionadas a um tópico consiste nos atores e na interação com aquele tema, discussão ou conteúdo, e pode ser operacionalizado como o conjunto de usuários, postagens, comentários, avaliações ou *likes/dislikes* relacionados ao tópico em questão. Por exemplo, um tópico pode ser representado por uma palavra-chave, como "#ukraine", o que, neste caso, as entidades relacionadas podem

consistir nos *tweets* que contêm aquela *hashtag* ou palavras relacionadas, como "#kyiv" e "#stoprussianaggression". Embora vamos discutir neste trabalho tópicos polarizadores de maneira textual, em princípio, os tópicos podem possuir diversas formas, uma vez que eles representem interações antagônicas entre usuários em torno de um tópico e isso pode surgir através de, por exemplo, mídias em vídeos ou fotos. A Seção 3.2 detalha as diferentes maneiras de realizar a extração e processamento destes dados.

Viés de entidades dos dados. Em uma segunda etapa do processo metodológico, unidades de informação referentes a um tópico de discussão são processadas a fim de se definir um posicionamento (referente aos polos). O objetivo aqui é particionar o conjunto de entidades em três conjuntos disjuntos: i) O conjunto de entidades que suportam o tópico em questão; ii) aqueles que são contrários ao tópico em questão e iii) aqueles que são neutros ao tópico em questão. Lembre que isso pode ser feito tanto de maneira discreta quanto em uma escala contínua. Em termos didáticos, o resultado desta etapa responde a seguinte questão: "assumindo que as entidades se dividem em dois conjuntos opostos de acordo com seu posicionamento com relação ao tópico, quais são esses conjuntos?". A Seção 3.3 detalha as diferentes técnicas para realizar o particionamento das atividades de um tópico.

Polarização de grupo. O terceiro estágio trabalha com os dados pré-processados já contendo a informação do viés de suas unidades. Intuitivamente, a polarização de um tópico que, a depender da técnica empregada, pode ser quantificada, expressa o quão bem separadas as duas partições estão, isto é, o quanto aqueles dois grupos divergem entre si. Apresentaremos diversas métricas para capturar a polarização, incluindo algumas baseadas em representações em baixa dimensão, estatística e teoria dos grafos. A Seção 3.4 apresenta maiores detalhes sobre essas técnicas.

Análise de polarização. A última etapa da metodologia adotada tem por objetivo realizar a análise dos dados e extração de conhecimentos a partir das etapas anteriores. Os tipos de análises e conclusões podem variar bastante a depender do tópico a ser analisado e do objetivo almejado. Alguns temas que podem ser explorados são o impacto das redes sociais na polarização encontrada na sociedade, análises do movimento da polarização ao longo do tempo, dentre outros tantos encontrados na literatura. A Seção 3.5 apresenta uma lista não exaustiva dos tópicos encontrados na literatura para que o leitor utilize como inspiração para seus trabalhos na área.

3.2. Dados no contexto de Polarização

Para derivar informações de polarização a partir de dados dados, é necessário que, primeiro, algumas medidas sejam tomadas. Nesta seção, serão destacadas as principais características e desafios encontrados ao lidar com dados relativos à polarização. Para tanto, serão apresentadas técnicas ou abordagens utilizadas para modelar e pré-processar esses dados. Ademais, será feito uma ligação entre o uso e a extração de informações sobre polarização, as quais são objeto de discussão do restante do capítulo.

3.2.1. A gênesis dos dados sobre polarização

A gênese dos dados, no contexto do estudo de polarização, é de pessoas que expressam seus pensamentos, desejos e/ou opiniões em diversos veículos propagadores de informa-

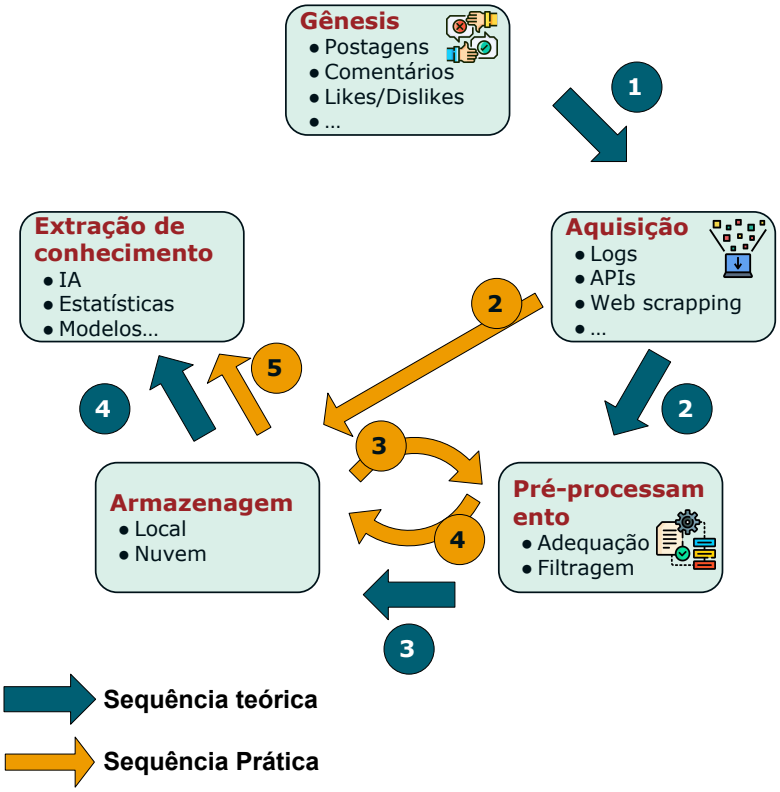


Figura 3.4: Etapas para extração da Polarização a partir de dados brutos

ção, tais como rodas de conversas, veículos de comunicação (TV, Rádio, blogs, jornais) e mídias social. Atualmente, essas últimas estão entre os principais veículos onde pessoas se expressam e propagam informações, normalmente na forma de postagens, comentários, indicação de aproximação através de *likes*, afastamento com *dislikes*, entre outras possibilidades.

Embora as mídias sociais não sejam os únicos veículos onde expressamos nossas opiniões, elas estão sendo massivamente usadas por pessoas como local de expressão [Gokcekus et al., 2021; Milroy and Llamas, 2013; Mitchell, 1974]. Pode-se ainda considerar, sem perda de generalidade, que as mídias sociais têm tido uma capacidade significativa de propiciar conhecimentos sobre posicionamentos de seus inscritos[6], bem como de grupos [Barros et al., 2021; Ferreira et al., 2021; Küçük and Can, 2020].

No contexto de coleta de dados em polarização, será considerado, neste trabalho, mídias sociais como "agregadores de dados de polarização", muito embora essa não seja sua finalidade. Ademais, destacamos que o conteúdo aqui apresentado pode ser extrapolado para outros domínios além das mídias sociais.

[6]Aqui estamos nos referindo a pessoas comuns, mas pode-se extrapolar para entidades genéricas, por exemplo, um conteúdo em vídeo/imagem ou outras quaisquer.

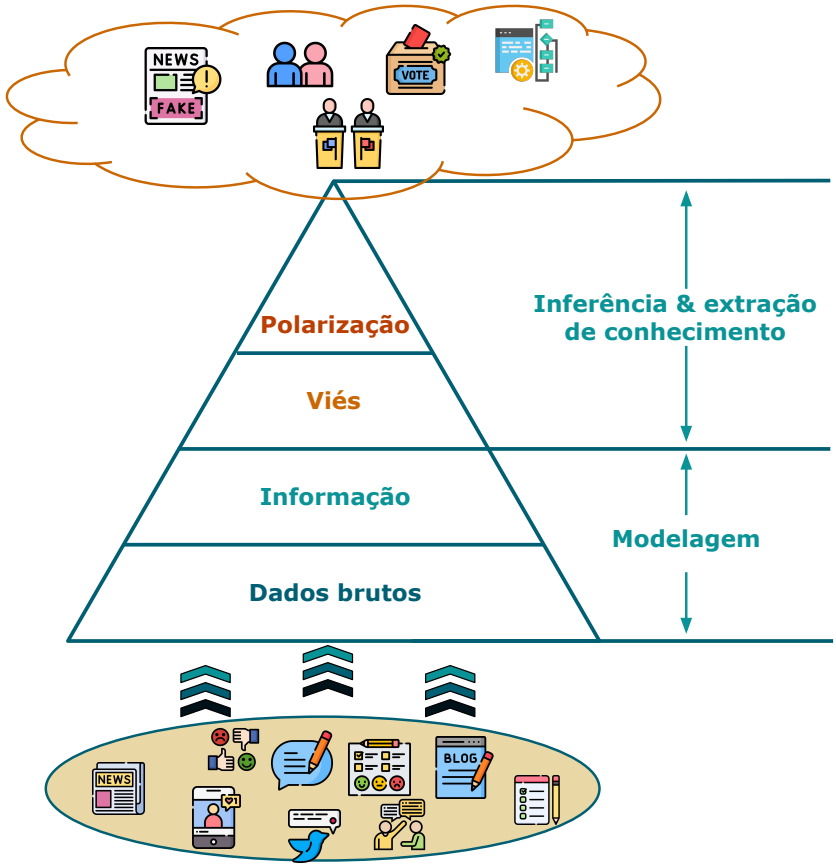


Figura 3.5: Hierarquia dos níveis de conhecimento sobre polarização a partir dos dados brutos

3.2.2. Desafios a partir dos dados

O principal problema aqui abordado é o de extração de conhecimento sobre polarização a partir de dados. Essa extração refere-se ao processo de modelar e analisar dados a fim de operacionalizar a inferência sobre posicionamentos de entidades individualmente ou em grupo. Por exemplo, em Küçük and Can [2020], os autores apresentam diferentes abordagens computacionais para indicar posicionamentos de *postagens individuais* em redes sociais ou textos comuns (ex.: blogs). Já em Ferreira et al. [2021], os autores estão preocupados com o posicionamento de *grupos de indivíduos* que interagem com um tópico de conversação em redes sociais.

Na Figura 3.4 são sumarizadas as etapas ideais e práticas para melhor entender o processo de extração de conhecimento. Iniciemos pelas etapas ideais (ou teóricas), as quais partem da aquisição/coleta de dados brutos sobre polarização (Etapa 1 em verde), passam pelo pré-processamento e armazenagem (Etapas 2 e 3 em verde), até a extração de conhecimentos a partir dos dados brutos (Etapa 4 em verde). As etapas intermediárias de pré-processamento e armazenamento são fundamentais para que os dados estejam aptos a serem utilizados por técnicas de inferência utilizadas na extração de conhecimentos sobre polarização.

Uma outra forma complementar de visualizar o processo é a abordagem hierár-

quica centrada em níveis de conhecimento conforme ilustrada na Figura 3.5, inspirada no trabalho de Santos et al. [2016]. Nessa abordagem, a transformação dos dados brutos é vista como uma hierarquia em que seus níveis são divididos em dois momentos: i) modelagem; e ii) análise e inferência sobre polarização. Na modelagem dos dados, o objetivo é adicionar algum nível de semântica e padronização aos dados brutos, os quais tipicamente são não estruturados, de fontes geradoras diferentes, e ainda podem possuir formatos heterogêneos. Nesta direção, técnicas de pré-processamento como representação de dados, filtragem ou fusão de dados podem ser aplicadas [Ayed et al., 2015; Khaleghi et al., 2013; Santos et al., 2016]. Já a análise e inferência tem por meta aplicar interpretações visando delinear o contexto a partir das informações e bem como extrair conhecimento acerca da situação daquele dado que, neste trabalho, é o *posicionamento* perante os grupos que se deseja estudar.

Embora as etapas intermediárias, entre os dados brutos e a extração de conhecimento sobre polarização, sejam essenciais, elas não são realizadas seguindo a sequência ideal. O que acontece na prática é a sequência destacada em amarelo na Figura 3.4, especialmente do ponto de vista de desenvolvimento. Tipicamente obtém-se os dados brutos e ciclos de pré-processamento e armazenamentos são realizados (em uma espécie de tentativas e erros/acertos) para que então os dados sejam encaminhados à algoritmos e técnicas de extração de conhecimento. O restante da seção tece considerações acerca do fluxo de dados na prática e suas particularidades do ponto de vista introdutório.

3.2.3. Aquisição dos dados

A primeira etapa do processo aqui abordado é o de aquisição de dados. Como os dados sobre polarização surgem de formas distintas, naturalmente as formas de aquisição também serão. Por exemplo, dados como texto opinativos de blogs ou páginas web pessoais podem ser coletados através de *Web Scraping*, enquanto *posts* em redes sociais podem ser adquiridos através de *Application Programming Interfaces (APIs)* públicas disponibilizadas pelos proprietários da rede social. Além disso, há diferentes formas de aquisição de dados sobre polarização, como por exemplo, questionários/formulários, observação, entrevistas, histórias orais, dentre outras. Aqui, serão comentadas as abordagens automáticas *Web Scrapping* e via *APIs*, tipicamente utilizadas para coletar dados da Internet.

O processo de ***Web Scrapping***, em português "raspagem da web", é uma técnica utilizada para extrair conteúdos ou conjuntos de dados diretamente de páginas da web [Khder, 2021]. Esse processo de extração de dados pode ser feito de modo manual ou automático, o qual popularmente se chama de *bot* ou *scraper*. É possível realizar *Web Scrapping* de diversas formas, e dentre elas, três ferramentas gratuitas se destacam: i) O FrameWork Scrapy[7] e ii) Beautiful Soup[8]; iii) Selenium[9]. O primeiro foi originalmente projetado para *Web Scrapping*, embora possa ser utilizado para outras finalidades como, por exemplo, coletar dados de APIs. O segundo, Beautiful Soup, é uma biblioteca para Python que visa fornecer meios para analisar e "caminhar" (*tree travessal*) em páginas web, permitindo recuperar informações específicas contidas na página. Já o terceiro, Selenium, foi primariamente projetado para testar navegadores.

[7]https://scrapy.org/

[8]https://www.crummy.com/software/BeautifulSoup/

[9]https://www.selenium.dev/

No caso de ***APIs***, é comum que grandes portais web e, especialmente, as grandes redes sociais exponham APIs que permitam a integração e expansão dos seus conteúdos e serviços. Essas APIs geralmente seguem a arquitetura e orientações do padrão Web REpresentational State Transfere (REST)[10], que define uma interface comum (tipicamente humano-legível) para integração e acesso a recursos. Esse acesso é realizado através de Uniform Resource Identifiers (URIs)[11] (*endpoints* fornecidos pelo provedor de serviços). REST Web APIs são baseadas no protocolo HTTP e, portanto, se apoia em seus métodos como GET, POST, PUT, DELETE e OPTIONS. Uma questão associada ao uso das APIs REST é a limitação na quantidade de requisições possíveis. Essa limitação varia de provedor para provedor e de recurso para recurso. Ao utilizar essas ferramentas, dados brutos podem ser coletados, tais como reportagens, colunas, manchetes, opiniões, comentários, etc.

Redes sociais como Twitter, Facebook, Instagram, entre outras, expõem APIs para acesso a seus recursos [Batrinca and Treleaven, 2015]. Por isso, dados dessas redes sociais têm sido tão utilizados para o estudo sobre polarização [Arora et al., 2022; Tucker et al., 2018]. Através das APIs dessas redes sociais é possível coletar dados brutos sobre *posts*, *likes/dislikes*, a rede de amizade, tópicos/assuntos em alta, *hashtags*, etc.

3.2.4. Pré-processamento

As tarefas de modelagem e filtragem dos dados são essenciais para transformar dados brutos em informações úteis. Dados sobre polarização podem não possuir uma organização lógica e hierárquica, relacionamentos ou serem completamente desestruturados, o que dificulta sua manipulação. Neste sentido, o desafio da modelagem dos dados brutos é uma representação uniforme para manipular esses dados e garantir que estão seguindo formatos padrões interpretáveis. Já a tarefa de filtragem lida com a eliminação de dados indesejáveis visto que é comum que dados adquiridos possuam imperfeições, erros de inserções, informações irrelevantes, dentre outras, os quais podem eventualmente adicionar vieses prejudiciais às análises ou inferências [Garimella et al., 2018a; Lu et al., 2015; Pannucci and Wilkins, 2010]. A seguir, apresentamos algumas técnicas de representação dos dados e questões de filtragem úteis para a etapa de pré-processamento dos dados.

3.2.4.1. Modelagem dos dados

A seguir listamos, de modo não exaustivo, técnicas para representações conceituais de dados. Salientamos que a aplicabilidade de cada representação pode variar a depender das finalidades da aplicação. Por esse motivo, destacamos os prós e contras de cada técnica. Estudos mais aprofundados podem ser encontrados em Bettini et al. [2010]; Garimella et al. [2018b]; Santos et al. [2016].

Modelagem dos dados: a modelagem dos dados lida com o problema de representar informação (organizada) a partir de dados brutos. Apresentamos, a seguir, uma lista não exaustiva de técnicas frequentemente utilizadas para modelar dados no contexto de polarização.

[10]https://restfulapi.net/

[11]RFC 3986: Uniform Resource Identifier (URI)

- Chave-valor (*key-value*) é uma estratégia para armazenar e recuperar arranjos associativos de informação (ex., dicionários). Nesta abordagem, pares de informação relacionadas (chave e valor) são os elementos básicos. Por exemplo, a chave pode ser o código do usuário em uma rede social e o valor uma lista de mensagens postadas por esse usuário.

- O *Markup Schema*[12] é outra estratégia que visa propor e manter esquemas para dados estruturados na Internet. Esses esquemas são um conjunto de 'tipos' (ex.: *CreativeWork, Book, Movie, Organization, Person*, etc) e cada tipo tem um conjunto de propriedades definidas (ex.: tipo *Book: bookEdition, bookFormat, illustrator*, et). O interessante é que esses esquemas podem ser utilizados em combinação com diferentes formatos de codificação como, por exemplo, RDF, Microdata, JSON-LD.

- A modelagem por *grafos* é bastante comum nos estudos sobre polarização [Belcastro et al., 2020; Coletto et al., 2017; Conover et al., 2011; Garimella et al., 2018a]. Tipicamente redes podem ser construídas a partir de relações entre pessoas (ex.: amizades em redes sociais), interações com postagens, comentários, *likes*, dentre outras. É comum que dados sejam modelados em grafos usando as seguintes abordagens clássicas: lista de arestas, lista de adjacência e matriz de adjacência. No contexto de polarização, cada nó do grafo é normalmente associado com um atributo que denota o posicionamento (ex: a favor) daquele nó em relação ao objeto de estudo (ex: legalização das drogas).

Além dessas três abordagens, outras também são possíveis, como representação usando ontologias, baseadas em objetos, baseadas em lógicas, entre outras [Bettini et al., 2010; Santos et al., 2016].

3.2.4.2. Filtragem dos dados:

O foco principal da filtragem de dados é a eliminação de dados não relevantes visando melhorar a qualidade da informação e, consequentemente, a qualidade das análises e inferências. Muitas questões relacionadas a filtragem dos dados podem acontecer [Rettore et al., 2016]. A seguir apresentamos uma lista não exaustiva de problemas recorrentes em dados sobre polarização que podem requer alguma técnica de filtragem.

- Granularidade dos dados: é a medida no nível de detalhes do dados coletados. Por exemplo, em uma série temporal referente a polarização, a medida de granularidade pode ser baseada na frequência em que os dados são capturados. Se em alta frequência, então tem-se mais detalhes sobre o comportamento das entidades perante aos polos, se menos frequente, tem-se menos riqueza sobre os o comportamento das entidades em torno dos pontos de estudo. Esse aspecto dos dados é importante no contexto de polarização, pois a partir dele, pode-se realizar inferências sobre entidades individuais (ex.: polarização de um único usuário, uma postagem ou comentário, etc.) ou grupos de entidades (ex.: polarização dos usuários de uma rede

[12]https://schema.org/

social). Em [Barros et al., 2021; Ferreira et al., 2021], os autores apresentam series temporais de dados que mostram o movimento dos indivíduos ao longo do tempo em torno dos polos contrastantes.

- Dados vagos: ocorre em conjunto de dados brutos onde seus atributos não estão bem definidos. Atributos com definição aberta/livre, permitem que dados sejam associados de modos subjetivos como, por exemplo, textos de mensagens na rede social Twitter (*tweets*) podem ser usados como dados brutos sobre a opinião de um usuário da plataforma. O conteúdo textual desse *tweet* pode ser vago e não prover informações suficientes para que seja possível detectar o posicionamento polarizador. Por exemplo, no contexto de do tema polarizador legalização de drogas, ao coletarmos *tweets* de um usuário que só faz comentários sobre outros temas (ex.: comida), não será possível identificar o posicionamento deste usuário sobre o tema legalização das drogas, tornando o *tweet* vago, ou em outras palavras, com baixa precisão, para este tema polarizador. Essa característica de baixa precisão também aparece ao utilizarmos expressões vagas, por exemplo, no texto hipotético *"A população teve um aumento* ***"expressivo/considerável/em torno"*** *em no* ***'tema polarizador'***. Embora possamos extrair a informação de aumento, expressões como *expressivo, considerável* não significam nada, o que torna conclusões sobre essa informação vagas, isto é, com baixa precisão.

- Outliers (anomalias): outro aspecto que pode implicar na necessidade de filtragem dos dados é a presença de *outliers*, os quais são pontos de dados que tipicamente diferem significativamente de outras observações [Grubbs, 1969]. No contexto de dados sobre polarização, anomalias podem aparecer de diferentes formas, desde sua inserção gênesis até na sua coleta ou processamento. Um exemplo de anomalias no contexto de polarização são *bots* que produzem spam ao divulgar opiniões extremas sobre um tema polarizador com o intuito de divulgar um posicionamento.

  Esses dados podem, eventualmente, distorcer as análises ao adicionar viés indesejável.

- Dados incompletos: são observações que possuem um ou mais atributos sem valor. Intuitivamente, essas partes ausentes podem gerar inferências e análises incorretas e, portanto, esses dados com partes faltantes podem ser filtrados. Suponha por exemplo que em um estudo sobre polarização, deseja-se estratificar a polarização por faixas etárias, porém, o atributo idade pode não estar presente para todos os indivíduos. Desta forma, teremos uma análise com dados incompletos.

3.2.5. Armazenamento

Para que a grande quantidade de dados gerados sobre polarização possam ser posteriormente analisada e processada, a etapa de armazenamento faz-se essencial. Esse armazenamento aparece na literatura de duas formas principais: armazenamento local ou em plataformas na nuvem voltadas ao armazenamento e, em alguns casos, processamento dessa massa de dados.

Em ambas as abordagens de armazenamento (local ou nuvem) seria desejável que os dados, logo após coletados, fossem adequados a um modelo facilitando a consulta subsequente. Entretanto, o que acontece com frequência, é um salto da aquisição dos dados

brutos para armazenamento utilizando um modelo mais simples e genérico possível e, eventualmente, esses dados passam por ciclos de pré-processamento (adequação a modelo(s) e filtragem) e armazenagem conforme ilustrado na Figura 3.4. É importante notar que a escolha de um Sistema de Gerenciamento de Banco de Dados (SGBD) e esquemas de para armazenagem de dados são etapas importantes no processo, porém estão fora do escopo deste trabalho.

3.2.6. Extração de conhecimento sobre Polarização

Como pode ser observado na Figura 3.5, os pré-processamentos permitem que os dados brutos sejam transformados em informação relevante sobre polarização a partir de uma estruturação em um modelo de representação comum e operações de filtragem. No nível subsequente da hierarquia, o *viés* refere-se a extração de qualquer informação que pode ser utilizada para caracterizar um eventual posicionamento (viés) perante aos possíveis polos/grupos que uma única entidade (postagem, pessoa, comentário, notícia) pode ter. Já o último nível da hierarquia representa a polarização em si, que geralmente é derivada a partir de diversas informações sobre vieses, proporcionando conhecimento global sobre a temática que apresenta contrastes ou agrupamento de opiniões. A etapa de análise e inferência busca definir a polarização e os vieses a partir dos dados coletados.

No restante deste capítulo serão discutidas técnicas para caracterizar viés e polarização. Para tanto, técnicas de inteligência computacional, estatísticas ou modelos matemáticos serão utilizados.

3.3. Viés de entidades dos dados

Quando olhamos para interações em redes sociais, podemos realizar diferentes tipos de análises com relação ao teor do conteúdo submetido e dos agentes participantes. Esses estudos variam tanto com respeito ao tipo de informação que se deseja estudar quanto ao elemento que se pretende observar. Identificar se uma determinada notícia ou postagem contém conteúdo falso, por exemplo, é uma tarefa de extrema relevância nos tempos atuais. Também podemos verificar se um comentário específico é ofensivo, ou identificar se um determinado usuário é um robô (*bot*) ou uma pessoa real. De forma geral, nos referimos a essas características como o viés dos dados.

A princípio, o estudo do viés de dados envolve um vasto conjunto de problemas relacionados. Análise de sentimentos, extração de opiniões, detecção de ironia, classificação de notícias falsas, mineração de argumentos, dentre outros. Cada um desses problemas representa uma área de estudo com amplas possibilidades, e envolve seus próprios métodos, métricas e modelos. Para este mini-curso, nosso maior interesse está no viés representado pelo posicionamento ou estância de um comentário ou usuário em específico (vide Figura 3.2). A partir dessa informação, poderemos realizar análises mais complexas a respeito de como os conteúdos de uma rede social se relacionam e conflitam entre si, nos levando ao conceito de polaridade.

Esta seção irá explorar o conceito de posicionamento, definindo o problema de detecção de posicionamento, comparando com outros problemas relacionados, e citando como ele é abordado em diferentes trabalhos. Desta forma, a seção se encarrega de apresentar soluções e ferramentas que são capazes de classificar um *tweet* ou comentário, ou

um usuário como um todo, com relação ao seu *viés* (muitas vezes, viés político).

3.3.1. O problema de detecção de posicionamento

A detecção de posicionamento [ALDayel and Magdy, 2021; Küçük and Can, 2020] é uma tarefa que trata de identificar o posicionamento (estância, orientação, apoio) que uma entidade presente nos dados representa com relação a um ou mais alvos (proposições, temas, tópicos). De forma geral, um alvo pode ser qualquer tipo de tema ou assunto a respeito do qual um usuário pode se posicionar. Comumente, esses temas podem representar aspectos ideológicos, decisões políticas, organizações, ou até mesmo a indivíduos específicos.

Um posicionamento, para o tipo de problema tratado neste mini-curso, identifica principalmente se o interlocutor está "*A Favor*" ou "*Contra*" àquela determinada proposição. Muitas vezes também são consideradas opções adicionais, como "*Nenhum*" e/ou "*Neutro*" para representar conteúdos que não se posicionam claramente a favor ou contra o alvo em questão. Nesses casos, uma estância "*Neutra*" representa um conteúdo que especificamente não é unicamente a favor nem contra aquele tema, enquanto "*Nenhum*" pode indicar comentários ou usuários que não se posicionam de forma clara naquele assunto ou simplesmente tratam de assuntos não-relacionados. Assim, formalmente, podemos definir o posicionamento de uma determinada entidade com relação a um tópico pela Equação 1, apresentada anteriormente.

3.3.1.1. Tipo de entidade

Podemos tratar o problema de detecção de posicionamento com relação a diferentes tipos de entidades dos dados nos quais avaliamos as estâncias. Há dois principais níveis de entidades que podemos encontrar sendo utilizados na literatura: declaração e usuário.

Quando aplicamos a detecção de posicionamento sobre uma declaração, nosso objetivo é identificar a orientação descrita em um determinado texto individualmente. Essa é uma tarefa de processamento de linguagem natural, e pode envolver textos mais curtos (sentenças), de tamanho médio (parágrafos, comentários, tweets), até textos de tamanhos mais longos (artigos jornalísticos, relatos).

Alternativamente, podemos detectar posicionamento ao nível de usuário. Nesse tipo de tarefa, deseja-se descobrir se cada usuário é, considerando-se todo o seu comportamento na plataforma digital, a favor ou contra o tema alvo. Isso pode incluir não só os textos de comentários e postagens produzidas por aquele usuário, como também outras informações de seu perfil (idade, gênero, etc.), comunidades e conteúdos com o qual interage, e relações com outros usuários em sua rede social.

3.3.1.2. Tipo de alvo

Também podemos diferenciar os trabalhos de detecção de posicionamento de acordo com o tipo de tópico ao qual os dados se referem. Na definição mais básica, uma entidade expressa posicionamento com relação a um tema. Dessa forma, deve-se construir um classificador separado para cada tema que se deseja identificar no conjunto de dados. Mas

também pode ser analisado como uma entidade se posiciona com relação a diversos temas relacionados simultaneamente. Por exemplo, para um estudo sobre eleições presidenciais, podemos analisar tweets e tentar detectar os posicionamentos de cada um com relação a todos os candidatos possíveis simultaneamente. Isso se torna vantajoso em casos onde os conteúdos dos dados costumam colocar os alvos posicionados relativamente um ao outro.

Existe ainda um terceiro tipo de alvo para o problema, no qual, ao invés de um ou mais temas explícitos, busca-se verificar como comentários em notícias se posicionam com relação a alegações feitas nas notícias em si. Normalmente, nesse tipo de problema, o objetivo é detectar se comentários confirmam ou negam determinadas informações dadas nas notícias, e pode ser usado como base para prever sua veracidade e identificar rumores e notícias falsas.

3.3.1.3. Outros problemas relacionados

Em estudos relacionados a conteúdos de mídias sociais, há diversos aspectos a serem abordados. Dentre esses, alguns se destacam por serem relacionados, e muitas vezes confundidos com a detecção de posicionamento. Para clarificar as diferenças entre esses aspectos e análises, e evitar erros conceituais, vamos agora comparar alguns dos principais problemas relacionados

Um dos principais problemas que são comumente utilizados de forma similar à detecção de posicionamento é a análise de sentimento [Liu, 2010, 2012; Ravi and Ravi, 2015]. Há estudos que utilizam o sentimento expresso em conteúdos como forma de determinar o posicionamento [Li and Caragea, 2019], seja diretamente ou como representação secundária. No entanto há diferenças conceituais fundamentais entre posicionamento e sentimento. Análise de sentimento tem como objetivo determinar a polaridade das emoções expressas em um determinado conteúdo, enquanto a detecção de posicionamento pretende identificar se um conteúdo expressa um ponto de vista a favor ou contra determinados tópicos.

De forma geral, podemos dizer que o sentimento é uma informação extraída do conteúdo em sua forma pura, enquanto um posicionamento é colocado como uma relação entre o conteúdo e o tema alvo. Uma simples frase como “estou nervoso!”, por exemplo, indica uma polaridade de sentimento clara, enquanto não necessariamente indica um posicionamento explícito com relação a qualquer tema por si só. Já uma declaração como “estou feliz que esse filme não fez sucesso” indicaria um sentimento positivo ao mesmo tempo que representa um posicionamento negativo com relação ao filme em questão.

Considerando essas diferenças entre as definições dos dois conceitos, é importante destacar que o uso de métricas de sentimento como um único fator determinante para a consideração de posicionamento não é adequado [Aldayel and Magdy, 2019; Mohammad et al., 2017; Sen et al., 2020]. No entanto, isso não quer dizer que não haja relação e interação entre sentimento e posicionamento, ou que não seja possível utilizar os dois conceitos em conjunto para compor uma análise [Tachaiya et al., 2021].

Outros problemas relacionados à detecção de posicionamento, porém distintos, incluem: reconhecimento de emoções [Canales and Martínez-Barco, 2014; Sailunaz et al.,

2018], no qual são identificadas emoções presentes em um conteúdo dentre um conjunto de classes de emoções; detecção de controvérsia [Coletto et al., 2017], que busca medir e identificar tópicos controversos em um conteúdo; previsão de posicionamento [Darwish et al., 2018; Dong et al., 2017], que se preocupa em estimar como usuários (ou grupos de usuários) se posicionariam com relação a temas dos quais esse posicionamento não foi observado, ao invés de detectar um posicionamento já explícito no conteúdo.

3.3.2. Abordagens de detecção de posicionamento

Tendo em vista o contexto mais específico de detecção de posicionamento, conforme definido anteriormente, podemos explorar os diferentes métodos utilizados para essa tarefa. A seguir, apresentamos uma visão geral dos atributos comumente usados nos modelos, assim como os algoritmos aplicados.

3.3.2.1. Atributos

Para abordar a detecção de posicionamento, há uma multitude de sinais que podem ser utilizados, dependendo do tipo de dado estudado. Isso pode incluir desde atributos extraídos diretamente do conteúdo até características da rede social em si, passando por representações geradas por modelos e análises de escolha de vocabulários.

Considerando o conteúdo textual de uma declaração diretamente, há trabalhos que utilizam modelagens dos termos em *bag-of-words* ou n-gramas como um conjunto primário de atributos [Mohammad et al., 2017], assim como outros indicadores como pontuação e tamanho do texto [Kochkina et al., 2017; Lai et al., 2017]. Além de características extraídas diretamente do texto, também é possível considerar métricas como a polaridade de sentimento do conteúdo [Ebrahimi et al., 2016], ou representações do conteúdo em forma de tópicos latentes [Elfardy and Diab, 2016].

Também há trabalhos que se voltam para como o vocabulário difere entre grupos com posicionamentos opostos [Darwish et al., 2020]. Com esse tipo de análise, pode-se por exemplo detectar a perspectiva de certos usuários com relação a determinados temas ao se observar como se comunicam no geral [Beigman Klebanov et al., 2010].

Além disso, o uso de características e atributos retirados a partir da rede também pode ser efetivo. Alguns trabalhos, por exemplo, modelam representações com base no texto do conteúdo em conjunto com características das interações do usuário na rede [Li et al., 2018], assim como atributos indicando as conexões e interações entre usuários na rede [ALDayel and Magdy, 2021], ou outras características retiradas das plataformas como hashtags, re-tweets, URLs, e menções a outros usuários [Darwish et al., 2017; Hamidian and Diab, 2019].

3.3.2.2. Algoritmos

A partir dos atributos extraídos dos conteúdos ou usuários, os estudos de detecção de posicionamento empregam diferentes métodos de aprendizado de máquina para identificar as estâncias daqueles elementos com relação aos temas em questão.

Métodos supervisionados são frequentemente utilizados para esse problema. Por essa abordagem, os dados relativos aos conteúdos ou usuários são anotados de acordo com seus posicionamentos. Normalmente, essas bases de dados são anotadas por especialistas de acordo com rótulos como "A Favor", "Contra"e "Nenhum", como aquela apresentada na tarefa de detecção de posicionamento em *SemEval-2016* [Mohammad et al., 2016]. A partir desses dados rotulados, algoritmos são treinados para aprender os padrões que indicam cada posicionamento. Trabalhos como Elfardy and Diab [2016] e Li and Caragea [2019] exemplificam esse tipo de abordagem, o primeiro com uso de atributos léxicos e semânticos em uma SVM (*Support-Vector Machine*) e o segundo com métodos de aprendizado profundo com uma arquitetura GRU (*Gated Recurrent Unit*).

No entanto, como esses dados rotulados são custosos para se produzir e de difícil obtenção, abordagens semi-supervisionadas e não-supervisionadas também são propostas. O trabalho de Ferreira and Vlachos [2019], por exemplo, usa técnicas de transferência de aprendizado para reutilizar o conhecimento que o algoritmo aprendeu em uma base de dado como ponto de partida para a detecção de posicionamento em dados de outras fontes. Já Zhang et al. [2020] tomaram diversas bases de dado em conjunto para detectar posicionamentos de forma cruzada em tarefas voltadas para temas de sub-grupos diferentes.

Métodos totalmente não-supervisionado também vem sido propostos, primariamente gerando representações dos usuários e conteúdos e aplicando métodos de agrupamento sobre elas. Darwish et al. [2020], por exemplo, aplicaram uma técnica de agrupamento sobre tweets não-rotulados de diferentes tópicos como ponto inicial para anotação dos dados. Outros como Rashed et al. [2020] usaram representações distribuídas de tweets com agrupamento hierárquico para análise de polarização política.

De forma geral, as possibilidades de algoritmos e técnicas para tarefa de detecção de posicionamentos é bastante ampla. Trabalhos como o de Swami et al. [2018] e Tsakalidis et al. [2018] utilizam SVMs. Outros como Kucher et al. [2018] e Ferreira and Vlachos [2016] aplicam regressão logística. Abordando técnicas de aprendizado profundo, são encontradas também diversas possibilidades, incluindo redes LSTMs (*Long Short-Term Memory*) [Rajendran et al., 2018], CNNs (*Convolutional Neural Network*) [Hercig et al., 2017], além de trabalhos utilizando mecanismos de atenção, como aqueles aplicando modelos baseados BERT [Ghosh et al., 2019; Kawintiranon and Singh, 2021]. E de forma geral, o uso de técnicas de agrupamento de modelos também são efetivas [Liu et al., 2016; Siddiqua et al., 2019].

3.4. Métricas de polarização

Como vimos na seção anterior, calcular o viés de um usuário ou conteúdo está ligado a verificar o posicionamento daquela entidade. Por exemplo, um comentário a favor de um tópico ou um usuário que declarou sua posição política. Já a polarização é uma medida de um grupo de usuários (ou um conjunto de conteúdos). Aqui estamos interessados em verificar como uma população está distribuída em torno de um tópico específico. Por exemplo: Em uma universidade a reitora A está passando por uma crise de popularidade. Foi então realizada uma pesquisa que revelou o viés de cada aluno. Destes, 40% dos alunos acreditam que a reitora tem feito um bom mandato, 45% que ela faz um mandato

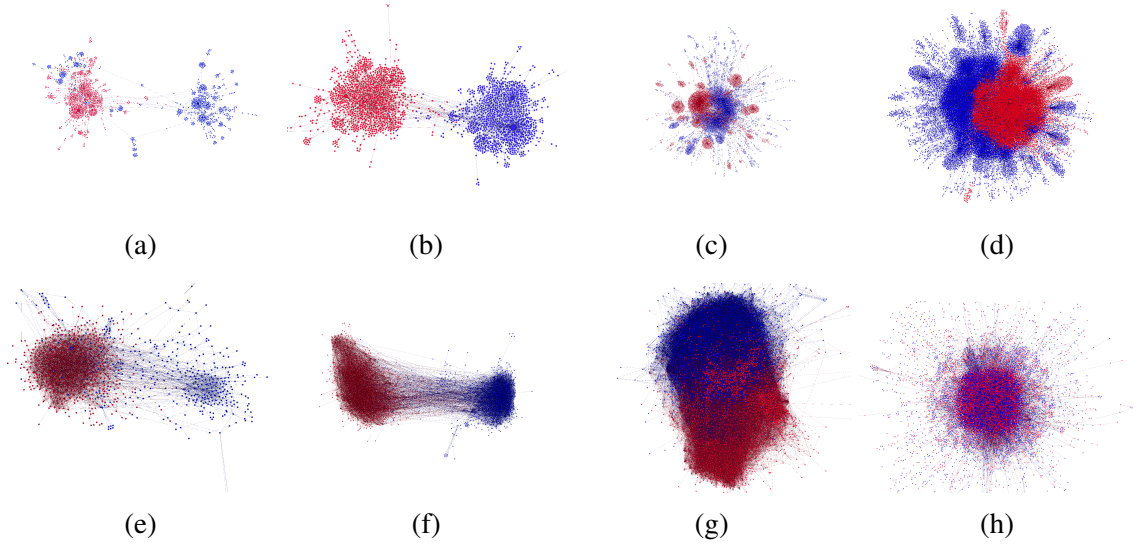


Figura 3.6: Exemplos de redes polarizadas (esquerda) e não polarizadas (direita). Os grafos superiores representam *retweets*, já os grafos inferiores representam *follow*. Mais informações sobre os dados coletados podem ser encontradas em Garimella et al. [2018b].

ruim e 15% ficaram neutros. Neste exemplo observamos que os alunos estão divididos em dois grandes grupos que discordam entre si (e um grupo menor que não tem uma opinião bem formada). Podemos então dizer que a universidade em questão está polarizada com relação ao mandato da reitora?

Outra forma de visualizar (e medir) a polarização é por meio do grafo da rede formado em torno de um tópico. Essa técnica é bastante utilizada no contexto das redes sociais, onde os usuários mantém ligações de amizade entre si, compartilhando e reagindo a postagens de outros usuários. Neste caso, cada usuário é um vértice do grafo e as arestas são formadas por meio das conexões da rede, como amizade e compartilhamento.

Na Figura 3.6 visualizamos alguns exemplos de redes (grafos) polarizadas e não polarizadas. As redes representadas em (a), (b), (e) e (f) são exemplos de redes polarizadas. Note que, em todas elas, temos dois grupos muito bem definidos e com poucas ligações (arestas) ligando seus vértices. Ainda na Figura 3.6, podemos observar exemplos de redes não polarizadas em (c), (d), (g) e (h). Observe que nos exemplos de redes não polarizadas temos os membros dos diferentes grupos com um número maior de ligações entre si, logo, menos isolados e com maior probabilidade de ter acesso a informações "do outro lado da rede".

Na literatura, não existe um consenso de como operacionalizar a quantificação da polarização de uma população. A maioria dos trabalhos podem ser caracterizados como estudos de caso, onde a polarização é identificada em bases de dados específicas e analisadas utilizando conhecimento do domínio (ex: lista de *hashtags* relacionadas a um evento político específico) [Garimella et al., 2018b]. Boa parte destas métricas foram desenvolvidas no contexto de redes sociais e são calculadas por meio de sua rede de comunicação [Conover et al., 2011; Garimella et al., 2018b; Guerra et al., 2013; Coletto et al., 2017; Matakos et al., 2017], outras são independentes da rede [Bakshy et al., 2015;

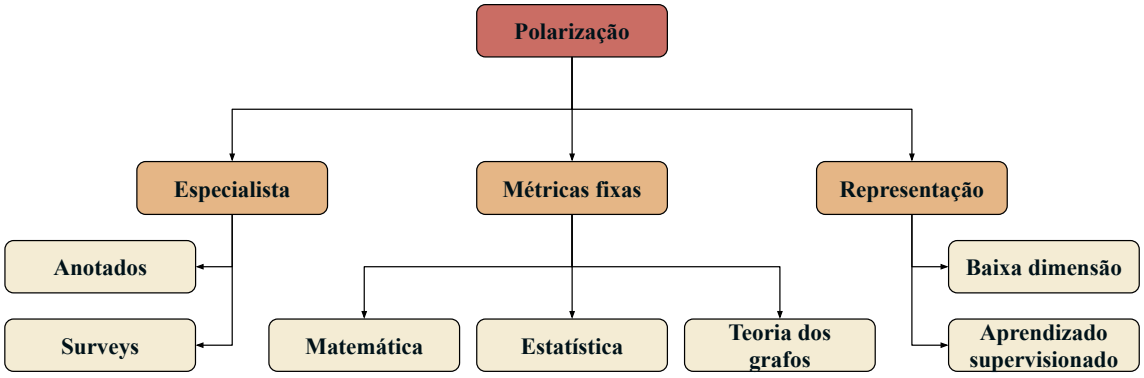


Figura 3.7: Taxonomia das métricas de polarização.

Belcastro et al., 2020; Roy and Goldwasser, 2020; Babaei et al., 2018; Morales et al., 2015; Vicario et al., 2019].

Uma métrica de polarização que não depende da estrutura da rede possui a vantagem de poder ser aplicada em cenários onde a estrutura da rede é desconhecida [Morales et al., 2015; Al-Ayyoub et al., 2018]. Tomemos como exemplo a caixa de comentários de uma famosa página de notícias. Neste caso, as opiniões ali deixadas tem como único ponto em comum a notícia em questão. Não temos informações sobre quais usuários possuem relações de amizade entre si ou ainda a qual viés político ele se alinha. No caso deste exemplo, precisamos de uma métrica de polarização que seja independente da estrutura da rede.

Em outros casos podemos usufruir de informações presentes na rede, como conexões de amizade ou de endosso (*retweet*, por exemplo). Os primeiros trabalhos analisaram a polarização em redes sociais utilizando a modularidade do grafo [Conover et al., 2011]. A modularidade de uma rede quantifica o quanto vértices conectam entre si formando comunidades densas, quando comparado a uma rede aleatória [Newman, 2006]. Outras análises procuram por padrões na rede, como os clusters de usuários [Garimella et al., 2018b], a fronteira entre os grupos [Guerra et al., 2013], entre outras estruturas [Coletto et al., 2017; Garimella et al., 2021]. Nestes casos estamos preocupados com as estruturas geradas na comunidade e como a comunidade está configurada.

Visando organizar as diferentes ferramentas utilizadas na literatura para identificar/quantificar tópicos polarizados, a Figura 3.7 apresenta uma proposta de taxonomia para as métricas de polarização. Dividimos as soluções em três tipos: (1) trabalhos que analisaram a polarização pela visão de um **especialista**, ao anotar os tópicos manualmente [Dori-Hacohen and Allan, 2015; Jang et al., 2016] ou por meio de questionários (*surveys*) [Ribeiro et al., 2019]. (2) Trabalhos que utilizaram **métricas fixas**, como fórmulas matemáticas [Al-Ayyoub et al., 2018; Belcastro et al., 2020], estatísticas [Jang and Allan, 2018; Morales et al., 2015] ou de teoria dos grafos [Garimella et al., 2018b; Guerra et al., 2013], para cálculo da polarização. E (3) trabalhos que utilizaram **representações** em baixa dimensionalidade [Waller and Anderson, 2021] e/ou algoritmos de aprendizado supervisionado [Roy and Goldwasser, 2020; Vicario et al., 2019] para identificação dos tópicos polarizados.

Tabela 3.1: Compilado de trabalhos relacionados em identificação de tópicos polarizados.

Trabalho	Conteúdo	Rede	Ident.	Quant.	Ferramenta
Al-Ayyoub et al. [2018]	✓		✓	✓	Matemática
Akhtar et al. [2019]	✓		✓	✓	Estatística
Babaei et al. [2018]	✓		✓	✓	Matemática
Belcastro et al. [2020]	✓		✓		Matemática
Choi et al. [2010]	✓		✓		Estatística
Dori-Hacohen and Allan [2015]	✓		✓		Anotado
Jang et al. [2016]	✓		✓		Anotado
Jang and Allan [2018]	✓		✓		Estatística
Klenner et al. [2014]	✓		✓		Aprend. superv.
Mejova et al. [2014]	✓		✓		Aprend. superv.
Morales et al. [2015]	✓		✓	✓	Estatística
Popescu and Pennacchiotti [2010]	✓		✓		Aprend. superv.
Ribeiro et al. [2019]	✓		✓		Survey
Roy and Goldwasser [2020]	✓		✓		Aprend. superv.
Tsytsarau et al. [2011]	✓		✓		Estatística
Vicario et al. [2019]	✓		✓	✓	Aprend. superv.
Waller and Anderson [2021]	✓		✓	✓	Baixa dimensão
Yang et al. [2017]	✓		✓	✓	Baixa dimensão
Akoglu [2014]		✓	✓		Estatística
Al Amin et al. [2017]		✓	✓		Aprend. superv.
Coletto et al. [2017]		✓	✓		Teoria dos grafos
Garimella et al. [2018b] (RWC)		✓	✓	✓	Teoria dos grafos
Garimella et al. [2018b] (BCC)		✓	✓	✓	Teoria dos grafos
Garimella et al. [2018b] (EC)		✓	✓	✓	Baixa dimensão
Garimella et al. [2021]		✓	✓		Teoria dos grafo
Gillani et al. [2018]		✓	✓		Estatística
Guerra et al. [2013]		✓	✓	✓	Teoria dos grafos
Shahrezaye et al. [2019]		✓	✓	✓	Teoria dos grafos
Tokita et al. [2021]		✓	✓	✓	Teoria dos grafos

Um compilado dos trabalhos relacionados pode ser visto na Tabela 3.1. Como vimos, os trabalhos são classificados entre aqueles que utilizam informação de conteúdo ou rede. Ainda, somente uma parcela dos trabalhos é capaz de quantificar o nível de polarização presente em um tópico.

Nas próximas seções iremos apresentar em detalhes algumas das principais métricas de polarização utilizadas na literatura. As seções 3.4.1 e 3.4.2 apresentam métricas de quantificação de polarização baseadas em conteúdo, isto é, não levam o grafo da rede em consideração. Já as métricas de polarização apresentadas nas seções 3.4.3, 3.4.4, 3.4.5 utilizam informações da rede para seu cálculo.

3.4.1. Utilizando análise de sentimentos para medir polarização

A primeira métrica de polarização que vamos abordar foi originalmente concebida para trabalhar com análise de sentimentos de tweets. Abordaremos aqui o trabalho de Al-

Ayyoub et al. [2018] que apresenta um conjunto de métricas matemáticas simples. Estas métricas podem ser utilizadas individualmente ou coletivamente na tarefa de análise de polarização, como mostraremos no decorrer da seção.

Razão da quantidade de tweets positivos e negativos (PN). Esta métrica se baseia na premissa de que, em um tópico polarizado, um dos grupos provavelmente utilizaria mensagens que demonstram sentimentos de aprovação ao tópico, e, o outro, mensagens de reprovação. Por consequência, esperamos observar um número de mensagens com sentimentos positivos no mesmo nível de mensagens negativas. Um maior valor para a razão PN representa uma maior grau de polarização. Também podemos utilizar a razão de tweets negativos e tweets positivos como uma métrica oposta a esta.

$$\text{PN} = \frac{|\text{tweets positivos}|}{|\text{tweets negativos}|} \tag{2}$$

Razão entre a quantidade de tweets positivos e negativos (RPN). A RPN é uma melhoria da PN vista anteriormente. Realizando sempre a razão entre o menor valor (tweets positivos ou negativos) e o maior valor, teremos como resultado um valor entre 0 e 1. Sendo que valores maiores representam uma maior polarização.

$$\text{RPN} = \frac{\min\{|\text{tweets positivos}|, |\text{tweets negativos}|\}}{\max\{|\text{tweets positivos}|, |\text{tweets negativos}|\}} \tag{3}$$

Razão entre a quantidade de tweets neutros e a quantidade de tweets positivos e negativos (NPN). Esta métrica se baseia na premissa de que, em um tópico polarizado, temos um pequeno número de mensagens neutras em comparação ao número de mensagens com viés claro. Isso aconteceria pois um número maior de usuários tenderia a demonstrar claramente sua aprovação ou desaprovação do tópico em questão. Nesta métrica, valores menores correspondem a uma maior polarização.

$$\text{NPN} = \frac{|\text{tweets neutros}|}{|\text{tweets positivos}| + |\text{tweets negativos}|} \tag{4}$$

Razão entre a soma dos tweets positivos e negativos com a quantidade total de tweets (PNT). Ainda com a premissa de que tópicos polarizantes obtêm um maior número de comentários com viés claro (positivo ou negativo), a métrica PNT calcula a razão entre a soma dos tweets positivos e negativos sobre o número total de tweets naquele tópico. Seu resultado varia entre 0 e 1, sendo que valores maiores representam uma maior polarização.

$$\text{PNT} = \frac{|\text{tweets positivos}| + |\text{tweets negativos}|}{|\text{tweets totais}|} \tag{5}$$

Métrica PN ponderada pela métrica PNT (PNPNT). Esta métrica é uma combinação entre a métrica PN e a métrica PNT. Ela calcula a razão entre os tweets positivos e negativos levando em consideração a razão entre os tweets com viés claro e o número total de tweets naquele tópico. Como a métrica PNT varia entre 0 e 1, o resultado da métrica PNPNT também varia entre 0 e 1, com valores maiores representando uma maior polarização.

$$\text{PNPNT} = \text{PN} \times \text{PNT} \tag{6}$$

Razão entre os valores de sentimentos positivos e negativos (RPNV). A RPNV é uma modificação da RPN vista anteriormente e leva em consideração os valores de viés calculados para cada tweet. Na RPNV calculamos a razão entre os valores de viés positivos e negativos e não o número de tweets. Valores próximos a 1 implicam que as opiniões contrárias possuem valores próximos, ou seja, que o valor total absoluto dos comentários positivos e negativos estão próximos. Os resultados da métrica RPNV variam entre 0 e 1, sendo que valores maiores representam uma maior polarização.

$$\text{RPNV} = \frac{\min\{\sum \text{valores positivos}, \sum \text{valores negativos}\}}{\max\{\sum \text{valores positivos}, \sum \text{valores negativos}\}} \tag{7}$$

Como podemos observar, as diversas métricas mostradas nesta seção capturam informações diferentes entre si. Cabendo ao leitor escolher a melhor para o seu trabalho ou ainda utilizar várias delas em conjunto. Como as métricas são fáceis de serem calculadas, elas são uma ótima opção para serem utilizadas como *features* para um algoritmo de aprendizado supervisionado. Podemos, por exemplo, elaborar um classificador automático de polarização em grupos de *WhatsApp* utilizando os sentimentos das mensagens que trafegam por ali. O trabalho de Al-Ayyoub et al. [2018] utiliza esse conjunto de métricas como entrada para uma *Support Vector Machine (SVM)* e, assim, classificar tópicos de discussão no Twitter. Além das métricas descritas nesta seção, o trabalho citado também utiliza a métrica *Dipole Moment (DM)* que será discutida na próxima seção.

Outro ponto importante é que as métricas podem ser facilmente adaptadas para outros cenários, como postagens em uma página de jornal ou votação de políticos na câmara de deputados. Basta a adaptação da métrica que calcula o viés de cada entidade. Por exemplo, podemos utilizar palavras-chaves ou ainda a opinião de um especialista.

3.4.2. Momento do dipolo elétrico

Nesta seção abordaremos a métrica introduzida pelo trabalho de Morales et al. [2015]. A métrica inspirada no momento do dipolo elétrico (em inglês, *Dipole Moment (DM)*) tem como objetivo capturar o quão dividido encontram-se os membros de uma população. Para isso parte-se da premissa de que uma população é perfeitamente polarizada se ela pode ser dividida em dois grupos de mesmo tamanho e com as opiniões de seus indivíduos concentradas nos extremos.

Sua inspiração vem da física com uma métrica que calcula a polarização das cargas de um sistema eletromagnético. Para isso ela calcula o grau de separação de cargas positivas e negativas que fazem parte do sistema. Um caso simples é onde temos somente duas cargas, uma negativa e uma positiva ($-q$ e $+q$), o momento do dipolo elétrico é proporcional à distância entre essas duas cargas. Esse caso é análogo a um cenário simples onde temos duas pessoas de opiniões contrárias acerca de um tópico. Logo, a polarização deste pequeno grupo pode ser calculada como o quão distante as opiniões destas duas pessoas se encontram.

Seja X uma variável aleatória que modela a distribuição do viés de uma população acerca de um tópico e X_i o viés de um usuário i de modo que $-1 \leq X_i \leq +1$. Então temos $p(X)$ como uma função de densidade da opinião dos usuários. Primeiramente, iremos calcular o tamanho das populações associadas a cada opinião (negativa e positiva). Seja

A^- a população com viés negativo ($X < 0$), calculamos seu tamanho como a área sob a curva da função de densidade $p(X)$ no intervalo $[-1,0)$ (equação 8). De maneira análoga, calculamos A^+ como a área sob a curva de $p(X)$ no intervalo $(0,+1]$ (equação 9).

$$A^- = \int_{-1}^{0} p(X)dX = P(X < 0), \tag{8}$$

$$A^+ = \int_{0}^{1} p(X)dX = P(X > 0) \tag{9}$$

De posse do tamanho dos grupos, estamos interessados em calcular a diferença absoluta entre eles. Que é facilmente calculada como podemos observar na equação 10. Esta fórmula dá como resultado $\Delta A = 0$ quando a população está perfeitamente dividida em dois grupos de tamanhos iguais ($A^- = A^+$). Por outro lado, $\Delta A = 1$ quando todos os elementos da população concordam entre si ($A^- = 1$ ou $A^+ = 1$).

$$\Delta A = |A^+ - A^-| = |P(X > 0) - P(X < 0)| \tag{10}$$

Em seguida, calculamos o quão distante estão as opiniões de ambos os grupos. Para isso, é calculado o centro de gravidade dos viéses negativos (gc^-) e positivos (gc^+), como podemos observar nas equações 11 e 12.

$$gc^- = \frac{\int_{-1}^{0} p(X)XdX}{\int_{-1}^{0} p(X)dX}, \tag{11}$$

$$gc^+ = \frac{\int_{0}^{1} p(X)XdX}{\int_{0}^{1} p(X)dX} \tag{12}$$

E então calculamos a distância entre as opiniões centrais de cada grupo como a diferença absoluta dos centros gravitacionais gc^- e gc^+, como podemos observar na equação 13. Esta fórmula dá como resultado $d = 0$ quando ambos os grupos concordam integralmente entre si. Por outro lado, teremos $d = 1$ quando ambos os grupos discordam entre si e suas opiniões se concentram nos extremos.

$$d = \frac{|gc^+ - gc^-|}{|X_{\max} - X_{\min}|} = \frac{|gc^+ - gc^-|}{2} \tag{13}$$

Por fim, calculamos o índice de polarização DM à partir dos valores calculados de ΔA e d(equações 10 e 13), como podemos observar na equação 14. Pela equação observamos que a métrica DM é inversamente proporcional à diferença absoluta entre as duas populações ΔA e diretamente proporcional à distância absoluta dos centros de gravidade d. Como o resultado de ΔA e d se encontram no intervalo $[0,1]$, o resultado de DM também se encontra no intervalo $[0,1]$.

$$DM = (1 - \Delta A)d \tag{14}$$

Teremos polarização máxima ($DM = 1$) quando a população estiver perfeitamente dividida ($\Delta A = 0$) e as opiniões discordantes estiverem nos extremos ($d = 1$). Por outro

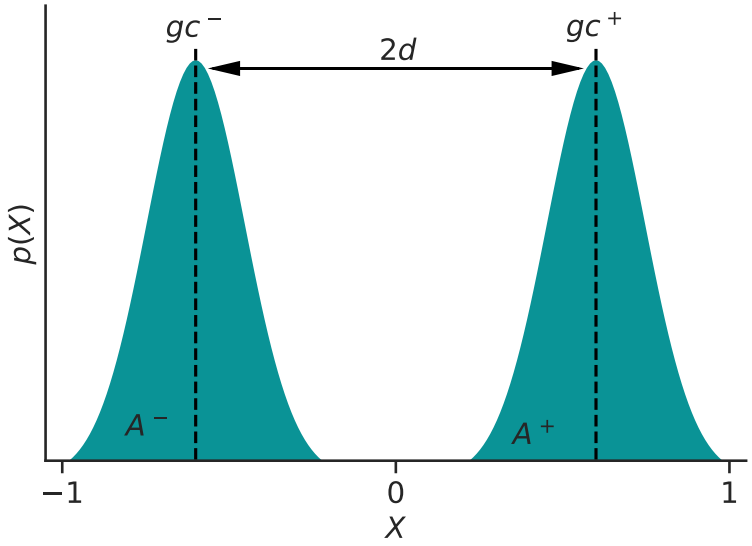


Figura 3.8: Representação da polarização e da métrica DM. Função de densidade de opinião. *A* representa a área associada a cada ideologia, *gc* representa o centro gravitacional de cada uma das opiniões e *d* representa a distância das opiniões.

lado, teremos polarização mínima ($DM = 0$) quando a opinião da população como um todo estiver concentrada em um único ponto, ou seja, quando não há discordância. A métrica poderá apresentar valores entre 0 e 1 quando as populações tiverem tamanhos desiguais ($0 < \Delta A < 1$) e/ou quando a distância entre os viéses das populações for menor que 1 ($0 < d < 1$). A Figura 3.8 ilustra os principais conceitos da métrica *DM*. Observamos a área que representa o tamanho de cada grupo com seu respectivo viés, assim como a distância entre cada centro de gravidade.

A métrica do momento do dipolo elétrico é muito útil quando o viés dos usuários (ou conteúdos) puderem ser quantificados (valores entre -1 e +1)e o grafo da rede de comunicação não for conhecido. A métrica é de fácil implementação e baixo custo computacional. Além disso, a função de densidade de opinião dos usuários pode ser uma ótima visualização da polarização da população. Uma análise desta função nos dá uma visão de como o tópico em questão foi recebido pela população e qual é a opinião predominante.

3.4.3. Conectividade na fronteira entre grupos antagônicos

A estrutura de uma rede social é afetada pelo contexto e comportamento dos usuários [Easley and Kleinberg, 2010]. Padrões de comportamento, como homofilia [McPherson et al., 2001], alteram a probabilidade que dois usuários se conectem. Em uma rede polarizada, é esperado encontrar padrões que representem a divisão de uma população. Um desses padrões é o antagonismo, isto é, conjuntos de usuários que não apresentarão laços (amizade, compartilhamento, etc) entre si.

Partindo dessa premissa, foram elaboradas métricas de polarização que utilizam informações da rede de modo a extrair informações topológicas do grafo da rede. O primeiro exemplo desse tipo de métrica de polarização é a métrica de *Conectividade de*

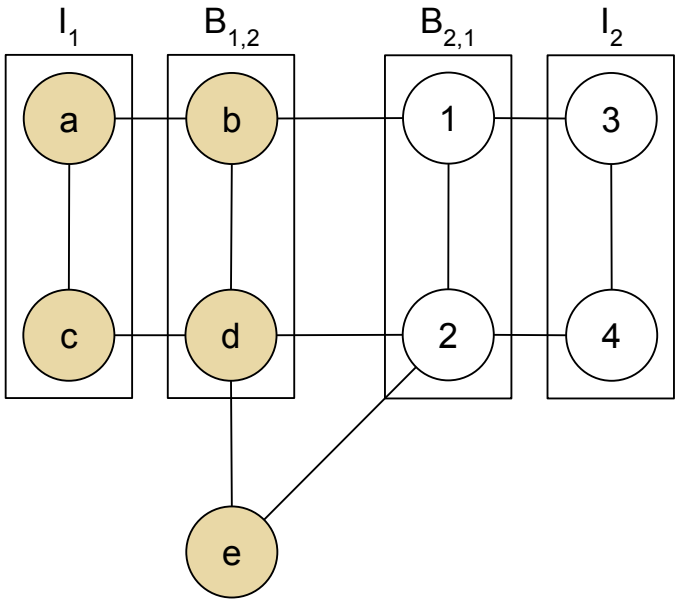


Figura 3.9: Exemplo de uma rede dividida em dois conjuntos G_1 (nós coloridos) e G_2 (nós brancos). Fonte: Guerra et al. [2013].

Fronteira (em inglês, *Boundary Connectivity* – BC) [Guerra et al., 2013]. Essa métrica foca sua análise nos nós que possuem alguma interação com nós do suposto grupo oposto, que aqui denominamos de nós de fronteira.

Seja $G = G_i \cup G_j$ o grafo que representa uma rede dividida em dois conjuntos G_i e G_j ($G_i \cap G_j = \emptyset$). Definimos fronteira de comunidade do grupo G_i, como o conjunto de nós $B_{i,j}$ que satisfaz duas condições:

1. $v \in G_i$ possui ao menos uma conexão com um nó do grupo G_j;
2. $v \in G_i$ possui ao menos uma conexão com um nó do grupo G_i que não possua conexão com os nós de G_j.

Na figura 3.9 temos um exemplo de uma rede dividida em dois grupos G_1 e G_2. Neste exemplo, temos as fronteiras $B_{1,2} = \{b,d\}$ e $B_{2,1} = \{1,2\}$. É importante ressaltar que o nó $c \notin B_{1,2}$ pois, apesar do nó c ter aresta com um nó do outro grupo (nó 2), ele não tem aresta com nenhum nó de G_1 que possua conexão com um nó em G_2. Ainda, nós em G_i que não pertencem à $B_{i,j}$ formam o conjunto de nós internos $I_i = G_i - B_{i,j}$. No exemplo, os nós internos são $I_1 = \{a,c,e\}$ e $I_2 = \{3,4\}$.

Vamos então nos ater aos nós que compõem a fronteira de modo a comparar o grau de preferência destes nós de se conectarem com nós internos ou com nós do outro conjunto. Voltando à Figura 3.9, vamos analisar as conexões do exemplo partindo do nó b. O nó b possui grau três e suas arestas são:

1. $(b,1)$ é uma aresta externa (liga nós de fronteiras opostas).
2. (b,a) é uma aresta interna (liga nós da fronteira a nós internos).

3. (b,d) não é nem uma aresta interna nem externa.

Olhando para as conexões do nó b, ele não nos parece apresentar nenhum tipo de antagonismo, uma vez que se conecta a uma aresta interna mas também a uma aresta externa. Essa mesma análise pode ser extrapolada para os demais nós da fronteira (nós d, 1 e 2). Logo, baseado nos nós da fronteira da rede, o exemplo da Figura 3.9 não possui polarização.

A equação 15 define a métrica *BC* que leva em consideração as escolhas que nós de $B_{i,j}$ fazem ao se conectar com nós de I_i ou $B_{j,i}$. Para cada nó v pertencente à fronteira B ele calcula a razão entre o número de arestas internas que ele possui – $d_i(v)$ – com o número de suas arestas externas – $d_b(v)$ – somadas ao número de suas arestas internas. Essa razão é comparada com a hipótese nula de que cada nó da fronteira tem a mesma probabilidade de possuir arestas com nós internos e nós externos.

$$BC = \frac{1}{|B|} \sum_{v \in B} \left[\frac{d_i(v)}{d_b(v) + d_i(v)} - 0.5 \right] \tag{15}$$

A métrica *BC* poderá apresentar valores entre $-1/2$ e $+1/2$. Um valor *BC* menor do que 0 indica não somente a falta de polarização mas também que um nó na fronteira é mais provável de se conectar com nós do grupo oposto. No exemplo da figura 3.9 temos $BC = 0$ uma vez que os nós da fronteira possuem o mesmo número de arestas com nós internos e com nós do grupo oposto.

A métrica BC apresenta a vantagem de utilizar informações de interação e comunicação entre os usuários, diferentemente das métricas vistas até agora. Além disso, seu foco na fronteira entre os grupos, mensura a interação entre os antagônicos, ou seja, a métrica nos traz uma visão sobre o nível de troca de informações entre os diferentes grupos.

3.4.4. Centralidade do corte da rede

Nesta seção iremos abordar a métrica de centralidade do corte da rede (em inglês, *Betweeness Centrality Controversy score* – BCC) [Garimella et al., 2018b]. Essa métrica vai analisar o conjunto de arestas presentes no corte que se forma ao particionar um grafo em dois grupos opostos X e Y. Para isso, utilizamos a noção de centralidade de arestas (em inglês, *edge betweeness*) e como a centralidade do corte difere das demais arestas.

A centralidade de uma aresta e é calculada pelo que chamamos de "intermediação" (em inglês, *shortest-path betweeness centrality* – $bc(e)$) [Brandes, 2008]. A equação 16 define a intermediação de uma aresta e, onde $\sigma_{s,t}$ é o menor caminho entre s e t, e $\sigma_{s,t}(e)$ é o menor caminho que necessariamente passa pelo vértice e.

$$bc(e) = \sum_{s \neq t \in V} \frac{\sigma_{s,t}(e)}{\sigma_{s,t}}, \tag{16}$$

Consideremos um grafo $G = (V,E)$ polarizado com grupos X e Y opostos e bem definidos ($X \cup Y = G$ e $X \cap Y = \emptyset$). Neste caso, o conjunto de arestas $C \subseteq E$ do corte $C = (X,Y)$ agiria como uma "ponte" levando informações entre os grupos. Fica então

fácil imaginar que o caminho mínimo entre pares de vértices de grupos opostos devem conter alguma aresta do conte C. O que leva a valores altos de intermediação (eq. 16) das arestas em C. Por outro lado, quando pensamos em um grafo onde os grupos não estejam fortemente separados, entendemos que existirão outras arestas pelas quais a informação pode passar. O que leva a valores de intermediação das arestas em C que sejam similares ao restante das arestas do grafo.

Vamos então transformar essa ideia em uma métrica que compara a distribuição do cálculo de centralidade das arestas do corte com a centralidade das demais arestas do grafo. Para isso é computada a divergência de Kullback-Leibler (d_{KL}) [Thomas and Joy, 2006] entre a distribuição de centralidade dos dois conjuntos de arestas. A divergência de KL é uma medida de distância entre duas distribuições de probabilidade (mais detalhes estão fora do escopo deste texto). A métrica *BCC* pode ser vista na equação 17 e é calculada como a inversa da divergência KL.

$$BCC = 1 - e^{-d_{KL}}, \tag{17}$$

O valor de *BCC* estará mais perto de 1 para tópicos polarizados e perto de 0 para tópicos não polarizados.

3.4.5. Random Walk Controversy (RWC)

A última métrica de polarização que iremos abordar almeja mensurar a facilidade do acesso da informação por usuários de grupos opostos. Se supormos uma rede polarizada com poucas ligações entre os grupos, o acesso ao grupo oposto será dificultado. Ao contrário, em uma rede não polarizada temos um número maior de arestas ligando membros de grupos distintos, e um maior tráfego de informação entre os grupos.

Utilizando caminhamento aleatório, a métrica *Random Walk Controversy* (RWC) [Garimella et al., 2018b] pretende calcular a probabilidade de um usuário acessar informações do grupo oposto. Considerando dois usuários que irão caminhar aleatoriamente sobre a rede, a métrica RWC (eq. 18) é definida como a diferença de probabilidade de que ambos terminem no grupo em que começaram e a probabilidade de que ambos terminem em grupos opostos aos quais começaram.

$$RWC = P_{XX}P_{YY} - P_{XY}P_{YX}, \tag{18}$$

onde $P_{AB}, A, B \in \{X, Y\}$ é a seguinte probabilidade condicional:

$$P_{AB} = P[\text{começou na partição A}|\text{terminou na partição B}]. \tag{19}$$

O resultado da métrica RWC ficará próximo a 1 quando a probabilidade de cruzar os grupos é baixa, o que implica uma alta polarização. Por outro lado, o resultado da métrica ficará próximo a 0 quando a probabilidade de cruzar os grupos é comparável a de terminar do mesmo lado, o que implica uma baixa polarização.

3.5. Análises sobre polarização

Chegamos no último estágio da metodologia que é a análise da polarização à partir dos dados coletados e processados. Como já discutido, a polarização é uma métrica amplamente utilizada em ambientes políticos, mas outros cenários também são possíveis, como

esportes [Guerra et al., 2013], discurso de ódio [Almerekhi et al., 2020], desinformação [Vicario et al., 2019; Watts et al., 2021], entre outros.

Logo, a análise da polarização deve levar em consideração as particularidades das áreas e os objetivos a serem alcançados. Nesse sentido, esta seção discute alguns dos tópicos de pesquisa encontrados na literatura. A seguir veremos uma lista dos tópicos abordados nesta seção.

1. *Baselines* para validação e comparação de métricas de polarização.
2. Utilização de polarização como *features* em áreas relacionadas, como hate-speech, toxicidade, fake-news.
3. Análises do impacto nas redes sociais de fatos que acontecem na vida real, por exemplo a morte de uma celebridade ou a corrida eleitoral em um país.
4. Previsão de resultados de eleições.
5. Análises de comportamento dos usuários em torno da polarização.
6. Impacto (ou influência) dos algoritmos das mídias sociais na sua polarização.
7. Abusos do uso da polarização de forma deliberada para ganho próprio.
8. Ferramentas e soluções para conter a polarização.

3.5.1. *Baselines* das métricas de polarização

Dada a diversidade de técnicas e métricas propostas na literatura para identificar polarização, alguns trabalhos direcionaram seus esforços em validá-las e compará-las. Como vimos neste curso (seção 3.4), alguns trabalhos focaram em métricas de polarização que utilizam o grafo da rede [Guerra et al., 2013; Garimella et al., 2018b]. Outros vão comparar as métricas baseadas em conteúdo [Al-Ayyoub et al., 2018; Morales et al., 2015].

Alguns trabalhos exploraram métricas que são independentementes do conteúdo [Garimella et al., 2018b; Guerra et al., 2013]. Outras métricas ainda demandam de um especialista [Jang et al., 2016; Ribeiro et al., 2019]. De toda forma, estudos nesta área se mostraram importantes pois ainda não existe um consenso sobre como medir polarização.

3.5.2. Polarização como *features*

As métricas de polarização não necessariamente devem ser vistam como um fim em si mesmas. Podemos explorar seu uso como uma ferramenta de apoio em áreas correlatas, como classificação discurso de ódio [Akhtar et al., 2019], toxicidade [Guimaraes and Weikum, 2021] e fake-news [Vicario et al., 2019]. É fácil pensarmos que em tópicos com presença de discurso de ódio, por exemplo, provavelmente teremos uma população dividida (polarizada). Tomemos uma discussão com presença de discurso de ódio ocorrendo nos comentários de uma notícia envolvendo racismo. É esperado que a população em torno dessa discussão se divida entre aqueles que atacam com ódio e aqueles que rechaçam o racismo e a intolerância.

De modo geral, os trabalhos vão percorrer suas bases de dados em busca de tópicos identificados como polarizados. E então se debruçar sobre estes tópicos de várias formas. Por exemplo, o nível de discordância (polarização) entre os especialistas que classificam uma base de dados pode ser utilizada juntamente com o conjunto de treino e gerar melhores classificadores [Akhtar et al., 2019].

Ainda, métricas de polarização podem ser utilizadas para melhorar previsões do aparecimento de mensagens tóxicas em uma conversa [Guimaraes and Weikum, 2021] ou ainda para prever prováveis tópicos alvos de notícias falsas [Vicario et al., 2019]. Todos estes são exemplos de classificadores onde a polarização não é o único aspecto a ser considerado, porém utilizar essas informação trouxe ganho aos classificadores.

3.5.3. Impactos de eventos nas mídias sociais

Outro ponto de interesse é a analise da comunicação nas mídias sociais atreladas a eventos do mundo real. Na literatura, encontramos diversos exemplos com eventos importantes, como a análise da polarização quando da morte do Hugo Cháves [Morales et al., 2015] e, em consequência, a crise política instaurada na Venezuela em 2019 [Horawalavithana et al., 2021]; o processo de Impeachment da presidenta Dilma Roussef no Brasil [Moreira et al., 2020], a corrida eleitoral de Donald Trump em 2016 [Yang et al., 2017], dentre outros.

Cada trabalho abordando como as mídias sociais reagiram aos eventos da vida real levando suas particularidades em consideração. Por exemplo, o trabalho de Moreira et al. [2020] realizou uma análise comparativa entre a polarização de dois segmentos da população: a "elite" (classe política) e a "massa" (população comum). Com essa particularidade, o cálculo do viés da "elite", foi realizado à partir das votações ocorridas na câmara dos deputados. Já o trabalho de Morales et al. [2015] se interessou em analisar a polarização dividindo a população em diferentes regiões geográficas.

3.5.4. Previsão de resultados de eleições

Como vimos, muitas métricas de polarização partem da classificação e quantificação de grupos antagônicos da população. Quando o tópico escolhido é aborto, por exemplo, temos uma ideia do tamanho da população pró-escolha e o tamanho da população pró-vida [Lu et al., 2015]. Quando o tópico é um jogo de futebol, quantificamos o tamanho da população que torce para cada um dos times [Guerra et al., 2013]. E, quando o tópico é uma corrida eleitoral, a análise da polarização em torno de um candidato nos mostrará o tamanho da população que o apoia.

No trabalho de Belcastro et al. [2020], por exemplo, os autores fizeram a classificação dos usuários do Twitter durante as eleições de 2016. Para isso, o viés dos usuários foi calculado como a razão entre o número de tweets compartilhados em apoio a cada candidato. Como resultado, eles obtiveram uma previsão da votação mais acurada do que as pesquisas políticas tradicionais. O mesmo resultado positivo foi observado com os dados da eleição de Donald Trump em 2016.

Poder analisar, e até prever, a preferencia dos usuários tem várias aplicações a depender do tópico escolhido. Uma empresa pode estar interessada em prever a recepção de um novo produto, ou um político, em saber o nível de aprovação da população a um

novo projeto, dentre outras aplicações.

3.5.5. Análise do comportamento dos usuários

Outras análises focam em um melhor entendimento do comportamento dos usuários em uma rede polarizada e quais são suas consequências. Os pontos a serem abordados podem variar como, por exemplo, uma caracterização da rede e dos seus usuários [Bakshy et al., 2015], um melhor entendimento do processo de homofilia e das câmaras de eco [Bright, 2017; Tokita et al., 2021], dos conteúdos que tem maior alcance em compartilhamento [Bakshy et al., 2015; Bright, 2017; Weld et al., 2021] ou tempo de permanência dos leitores em páginas com diferentes vieses [Garimella et al., 2021],

Ainda, alguns autores escolhem trabalhar com simulações das redes e do comportamento dos usuários. Essas simulações possibilitaram avaliar o impacto da polarização na estrutura da rede [Tokita et al., 2021]. E entender o papel das decisões dos usuários ou do algoritmo da rede social [Garimella et al., 2021; Valensise et al., 2022]. As diferentes particularidade entre as redes sociais, como o Twitter ou o Reddit, e seus impactos na polarização, também é um tópico a ser explorado [Weld et al., 2021].

O trabalho de Waller and Anderson [2021], por exemplo, fez uma análise de todas as comunidades do Reddit de forma a analisar a polarização política total da plataforma desde a sua criação. Dentre outras coisas, eles observaram que a polarização começou a aumentar durante o ano de 2016 com as eleições do Trump e não voltaram a diminuir. Ainda, eles observaram que a polarização se dá pelos novos usuários que entram na plataforma primariamente.

3.5.6. Influência dos algoritmos na polarização

Vimos na última seção que indícios apontam um aumento da polarização ao longo dos anos. Uma pergunta importante a ser respondida é se o advento da Internet juntamente com as redes sociais possuem sua parcela de responsabilidade.

Alguns autores exploraram essa questão, por exemplo, o trabalho de Kulshrestha et al. [2017] focou em calcular o viés do algoritmo de busca do Twitter. Para isso, ele calcula os vieses dos tweets que retornam com respostas a determinadas consultas. Ainda, o trabalho de Valensise et al. [2022], criou um modelo de simulação para entender o impacto da escolha dos usuários e do algoritmo das mídias sociais. Como resultado o trabalho concluiu que o algoritmo tem um grande papel na polarização dos usuários.

3.5.7. Abusos do uso da polarização

Outra investigação a ser realizada é a existência de elementos que utilizam a polarização de forma deliberada. O trabalho de Ribeiro et al. [2019] investigou uma série de anúncios políticos disparados por uma agência russa ao povo americano durante as eleições de 2016. O trabalho apontou que esses anúncios foram disparados no Facebook com o intuito de explorar a polarização da rede. Tais anúncios eram direcionados para perfis específicos de usuários, possuinudo um alcance 10 vezes maior do que a média de anúncios na plataforma. Neste caso em particular, os atacantes procuraram perfis de usuários pertencentes a populações específicas (principalmente liberais e negros) com o objetivo de criar discórdia.

3.5.8. Contenção da polarização

Como vimos, existem alguns indícios de que a polarização vem aumentando. O leitor pode então estar se perguntando quais seriam as soluções existentes para mudar este cenário. Na literatura, alguns trabalhos apresentaram ferramentas e soluções para diminuir a polarização presente nas redes sociais. Como exemplos desses trabalhos, temos a recomendação de usuários e conteúdos com viés diferente [Gillani et al., 2018], a elaboração de *feeds* de notícias que não sejam enviesados [Babaei et al., 2018; Jang and Allan, 2018]. Ou ainda, alguns trabalhos propõe mudanças na rede de comunicação ao adicionar conexões entre antagônicos [Garimella et al., 2017a] ou adicionar novos nós na rede [Garimella et al., 2017b]. De modo geral, o objetivo dessas soluções é diminuir as câmaras de eco, promovendo o contato com visões opostas.

3.6. Prática: Covid-19 e Hidroxicloroquina (HCQ)

Entendemos que uma abordagem prática é de suma importância para a fixação dos conteúdos aprendidos. Nesta seção, apresentamos um exemplo prático completo desde a coleta até a medição e análise de polarização em rede social. O código fonte utilizado nesta seção podem ser acessados em: `https://github.com/brhott/webmedia2022-polarization`. Para este estudo prático utilizaremos ferramentas e APIs desenvolvidas na linguagem Python[13]. As ferramentas serão apresentadas no desenvolver da seção.

3.6.1. Base de dados

A primeira fase é a escolha e obtenção da base de dados a ser estudada. Neste exemplo, escolhemos analisar o tópico do uso do medicamento Hidroxicloroquina (HCQ) para o tratamento da Covid-19. A base de dados será composta por um conjunto de tweets acerca do tema. Neste caso não nos preocuparemos com os usuários, realizaremos uma análise de polarização baseada no conteúdo postado. Para coleta, utilizamos a ferramenta *Tweepy* [14] para acesso à API do Twitter [15].

O código de coleta de dados pode ser visualizado no Programa 3.1. Nas duas primeiras linhas importamos as bibliotecas Tweepy e *Pandas* [16] – Pandas é uma biblioteca para manipulação e análise de dados. Na linha 4, inicializamos a biblioteca *Tweepy* com uma chave de acesso denominada `Bearer_Token` – mais informações sobre instalação e configuração da biblioteca *Tweepy* podem ser encontradas em `tweepy.org`. Vamos buscar por 100 *tweets* que contenham as palavras `hydroxychloroquine`, `chloroquine` e `HCQ`; eliminando os *retweets* (linhas 5 e 6). E, com a linha 7, armazenamos o resultado da nossa consulta em um *Dataframe* Pandas.

Aprendemos como realizar uma consulta utilizando a biblioteca *Tweepy*. Porém, para o restante desta seção, vamos utilizar a base de dados disponibilizada no trabalho de Mutlu et al. [2020]. Esta base contém um total de 14.374 *tweets* sobre o uso da Hidroxicloroquina como medicamento para a covid-19. Os *tweets* foram coletados durante todo

[13] `https://python.org`
[14] `https://tweepy.org`
[15] `https://developer.twitter.com`
[16] `https://pandas.pydata.org`

```
1 import tweepy
2 import pandas as pd
3
4 client = tweepy.Client(Bearer_Token)
5 query = "hydroxychloroquine chloroquine HCQ -is:retweet"
6 tweets = client.search_recent_tweets(query=query, max_results=100)
7 df = pd.DataFrame(tweets.data).set_index('id')
```

Programa 3.1: Download de uma base de dados utilizando a API do Twitter

```
1 from nltk.sentiment.vader import SentimentIntensityAnalyzer
2 sid = SentimentIntensityAnalyzer() #inicializacao
3
4 scores = df['text'].apply(lambda text : sid.polarity_scores(text))
5 df['compound'] = scores.apply(lambda score : score['compound'])
```

Programa 3.2: Análise de sentimentos da base de dados utilizando a ferramenta *Vader*

o mês de Abril de 2020 e seus vieses foram classificados manualmente. Essa classificação foi feita com relação ao seguinte questionamento: "coroquina/hidroxicoroquina é a cura para o novo coronavírus?". Desses, utilizamos apenas 9.117 tweets que continuavam online, sendo 3.732 tweets classificados como negativos, 3.385, positivos, e, 2.000, neutros. Mais detalhes sobre a base de dados e sua coleta podem ser encontrados no trabalho original.

3.6.2. Viés dos tweets

A segunda fase da metodologia é a de cálculo dos vieses dos dados. Como foi dito, optamos por utilizar uma base de dados onde o viés de cada tweet fora anotado manualmente. Porém, vamos abordar também a análise de sentimentos destes tweets, da mesma forma que alguns trabalhos da literatura [Al-Ayyoub et al., 2018; Vicario et al., 2019]. Nesse sentido, vamos abordar a utilização da ferramenta *Vader* [17] para extrair o sentimento de cada tweet da base de dados.

O código no Programa 3.2 realiza a inicialização e aplicação do *Vader* no texto dos tweets coletados (`df['text']`). Os resultados calculados são armazenados em `df['compound']`. Segundo a documentação da ferramenta, os valores são interpretados da seguinte maneira: valores no intervalo $[-1.0, -0.05)$ denominam um tweet com sentimento negativo; valores no intervalo $[-0.05, 0.05]$, sentimento neutro; e valores no intervalo $(0.05, 1.0]$, sentimento positivo.

3.6.3. Polarização do grupo

A terceira faze compreende na escolha e aplicação das métricas de polarização. Como estamos trabalhando com o conteúdo dos tweets e não com a estrutura da rede, vamos implementar as métricas das seções 3.4.1 e 3.4.2 (Programa 3.3). As linhas 2-4 separam os tweets negativos dos positivos. As linhas 6-13 realizam alguns cálculos intermediários para as métricas, como, por exemplo, a contagem de tweets negativos ou o cálculo do

[17] `https://nltk.org`

```
1  # Separacao entre tweets positivos e negativos.
2  g  = df['compound']
3  gn = df[df['compound'] <= -0.05]['compound']
4  gp = df[df['compound'] >=  0.05]['compound']
5
6  A   = g.count()      # num de tweets totais
7  An  = gn.count()     # populacao de tweets negativos
8  Ap  = gp.count()     # populacao de tweets positivos
9  A0  = A - An - Ap    # populacao de tweets neutros
10 Sn  = abs(gn.sum()) # soma dos valores de sentimento positivos
11 Sp  = gp.sum()       # soma dos valores de sentimento negativos
12 gcp = gp.mean()      # centroide dos tweets positivos
13 gcn = gn.mean()      # centroide dos tweets negativos
14
15 PN    = Ap / An                    # metrica PN
16 RPN   = min(An, Ap) / max(An, Ap) # metrica RPN
17 NPN   = A0 / (An + Ap)             # metrica NPN
18 PNT   = (Ap + An) / A              # metrica PNT
19 PNPNT = PN * PNT                   # metrica PNPNT
20 RPNV  = min(Sn, Sp) / max(Sn, Sp) # metrica RPNV
21
22 dA  = abs(Ap - An) / A # diferenca do tamanho das populacoes
23 d   = (gcp - gcn) / 2  # distancia entre centroides
24 m   = (1 - dA) * d     # metrica do dipolo eletrico
```

Programa 3.3: Cálculo da polarização do grupo com implementações dos trabalhos de Al-Ayyoub et al. [2018] e Morales et al. [2015].

centroide dos tweets positivos. Por sua vez, as linhas 15-20 trazem as diversas métricas que abordamos na seção 3.4.1. Por fim, as linhas 22-24 são responsáveis pela métrica de polarização do dipolo elétrico abordada na seção 3.4.2.

O código apresentado nesta seção utilizou os dados de análise de sentimento calculados pelo *Vader*. Porém, é fácil observar que as linhas 2-4 podem ser facilmente adaptadas para carregar os valores de viés obtidos por outros meios. Na próxima seção também iremos apresentar os resultados de cálculo das métricas utilizando os vieses classificados manualmente (como visto na seção 3.6.1).

3.6.4. Análise da polarização

A última etapa consiste na análise e discussão dos resultados. A Figura 3.10 apresenta a distribuição tweets em torno de seus vieses e nos dá uma ideia do nível de polarização em torno do tópico em questão. A figura da esquerda nos mostra a distribuição dos sentimentos dos tweets, onde podemos observar que os conjuntos de tweets negativos e positivos possuem distribuições parecidas. Quando analisamos os vieses anotados manualmente (figura à direita), observamos que os conjuntos de tweets negativos e positivos possuem tamanhos similares. Pela visualização de ambos os histogramas, podemos esperar uma alta polarização dessa população (população dividida entre dois grupos de posições opostas).

Os resultados das métricas de polarização calculadas podem ser visualizadas na

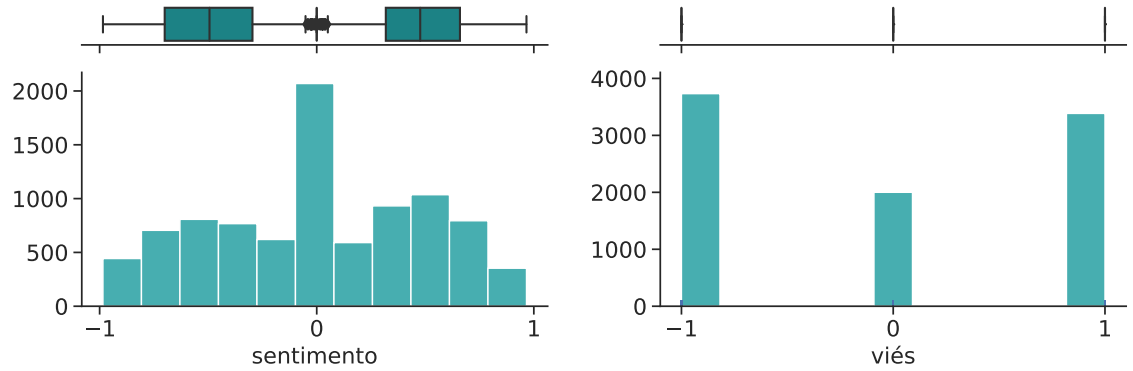


Figura 3.10: Histograma dos vieses dos tweets calculados com análise de sentimentos (esquerda) e manualmente (direita). Acima temos os boxplots de cada um dos grupos: negativo, neutro e positivo.

Tabela 3.2: Resultado das métricas de polarização utilizando vieses calculados por análise de sentimentos (S) e de maneira manual (M)

	PN	RPN	NPN	PNT	PNPNT	RPNV	DM
S	1.11	0.90	0.25	0.80	0.89	0.94	0.47
M	0.91	0.91	0.28	0.78	0.70	0.91	0.96

Tabela 3.2. Como previámos, a polarização encontrada foi alta na grande maioria das métricas. Uma adendo quanto a métrica NPN que é a única das métricas que inversamente proporcional à polarização. Uma segunda observação agora com relação à métrica do dipolo elétrico (DM) onde temos resultados diferentes entre as bases de dados (0.47 e 0.96). Essa métrica leva em consideração o grau de distanciamento das opiniões de cada um dos grupos. Nesse caso, o viés anotado manualmente com somente 3 valores – negativo (-1), neutro (0) e positivo (+1) – colocará a distância entre os grupos negativo e positivo como a maior possível. É importante ressaltar que não é prudente comparar os resultados com bases de dados processadas de maneira diferentes como a análise de sentimentos e a anotação manual.

De toda forma, o conjunto de métricas apresentadas na Tabela 3.2, juntamente com a visualização dos histogramas dos *tweets* apresentados na Figura 3.10 nos trazem valiosas informações acerca da polarização do tópico em questão. Vimos que os conjuntos antagônicos tem tamanhos próximos (PN, RPN), que o número de tweets neutros é baixo com relação aos tweets com vieses claros (NPN, PNT) e que a razão entre o somatório dos vieses de cada grupo fica próxima de 1 (RPNV), ou seja, que os grupos estão enviesados (distância até o centro) de maneira similar entre eles. Todas estas são características de uma população polarizada.

3.7. Considerações Finais

Identificar polarização nas redes sociais ainda é uma tarefa dependente do contexto. Neste capítulo, apresentamos uma revisão bibliográfica com o objetivo de contextualizar e apre-

sentar ao leitor o atual cenário da pesquisa em polarização. Para tanto, apresentamos os principais conceitos e definições da área além de prover ao leitor o ferramental necessário para elaborar suas próprias análises na área de polarização. Para isso, propusemos e apresentamos uma metodologia que passa pela coleta e processamento de dados no contexto da polarização; a classificação do viés das entidades presentes nos dados, com enfoque em texto; a escolha e aplicação de métricas de identificação e, a depender da técnica utilizada quantificação da polarização do tópico em questão; por fim, a análise e interpretação dos resultados da polarização.

Também foi proposta uma taxonomia das métricas e técnicas de polarização em redes sociais. Abordamos métricas elaboradas de diferentes meios, como métricas estatísticas e que utilizam técnicas de teoria dos grafos, o que evidencia a falta um consenso na literatura sobre como operacionalizar a identificação e a polarização de uma população. Nessa mesma linha, este capítulo apresentou diversos exemplos de trabalhos correlatos, seus resultados e suas implicações. Por fim, foi apresentado um exemplo prático de análise de polarização no contexto da pandemia Covid-19 e da discussão em torno do medicamento Hidroxicloroquina.

Agradecimentos. Este trabalho foi parcialmente financiado pelo CNPQ, FAPEMIG e FAPESP.

Referências

Akhtar, S., Basile, V., and Patti, V. (2019). A new measure of polarization in the annotation of hate speech. In *International Conference of the Italian Association for Artificial Intelligence*, pages 588–603. Springer.

Akoglu, L. (2014). Quantifying political polarity based on bipartite opinion networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, pages 2–11.

Al Amin, M. T., Aggarwal, C., Yao, S., Abdelzaher, T., and Kaplan, L. (2017). Unveiling polarization in social networks: A matrix factorization approach. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE.

Al-Ayyoub, M., Rabab'ah, A., Jararweh, Y., Al-Kabi, M. N., and Gupta, B. B. (2018). Studying the controversy in online crowds' interactions. *Applied Soft Computing*, 66:557–563.

Aldayel, A. and Magdy, W. (2019). Assessing sentiment of the expressed stance on social media. In Weber, I., Darwish, K. M., Wagner, C., Zagheni, E., Nelson, L., Aref, S., and Flöck, F., editors, *Social Informatics*, pages 277–286, Cham. Springer International Publishing.

ALDayel, A. and Magdy, W. (2021). Stance detection on social media: State of the art and trends. *Information Processing and Management*, 58(4).

Almerekhi, H., Kwak, H., Salminen, J., and Jansen, B. J. (2020). Are these comments triggering? predicting triggers of toxicity in online discussions. In *Proceedings of The Web Conference 2020*, pages 3033–3040.

Arora, S. D., Singh, G. P., Chakraborty, A., and Maity, M. (2022). Polarization and social media: A systematic review and research agenda. *Technological Forecasting and Social Change*, 183:121942.

Ayed, S. B., Trichili, H., and Alimi, A. M. (2015). Data fusion architectures: A survey and comparison. In *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 277–282. IEEE.

Babaei, M., Kulshrestha, J., Chakraborty, A., Benevenuto, F., Gummadi, K. P., and Weller, A. (2018). Purple feed: Identifying high consensus news posts on social media. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 10–16.

Bakshy, E., Messing, S., and Adamic, L. A. (2015). Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132.

Barros, M. F., Ferreira, C. H., Santos, B. P. d., Júnior, L. A., Mellia, M., and Almeida, J. M. (2021). Understanding mobility in networks: A node embedding approach. *arXiv preprint arXiv:2111.06161*.

Batrinca, B. and Treleaven, P. C. (2015). Social media analytics: a survey of techniques, tools and platforms. *Ai & Society*, 30(1):89–116.

Beigman Klebanov, B., Beigman, E., and Diermeier, D. (2010). Vocabulary choice as an indicator of perspective. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 253–257, Uppsala, Sweden. Association for Computational Linguistics.

Belcastro, L., Cantini, R., Marozzo, F., Talia, D., and Trunfio, P. (2020). Learning political polarization on social media using neural networks. *IEEE Access*, 8:47177–47187.

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., and Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and mobile computing*, 6(2):161–180.

Borah, A. and Singh, S. R. (2022). Investigating political polarization in India through the lens of Twitter. *Social Network Analysis and Mining*, 12(1):1–26.

Boxell, L., Gentzkow, M., and Shapiro, J. M. (2017). Greater internet use is not associated with faster growth in political polarization among us demographic groups. *Proceedings of the National Academy of Sciences*, 114(40):10612–10617.

Brandes, U. (2008). On variants of shortest-path betweenness centrality and their generic computation. *Social networks*, 30(2):136–145.

Bright, J. (2017). Explaining the emergence of echo chambers on social media: the role of ideology and extremism. *Available at SSRN 2839728*.

Canales, L. and Martínez-Barco, P. (2014). Emotion detection from text: A survey. In *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*, pages 37–43, Quito, Ecuador. Association for Computational Linguistics.

Choi, Y., Jung, Y., and Myaeng, S.-H. (2010). Identifying controversial issues and their sub-topics in news articles. In *Pacific-Asia Workshop on Intelligence and Security Informatics*, pages 140–153. Springer.

Coletto, M., Garimella, K., Gionis, A., and Lucchese, C. (2017). Automatic controversy detection in social media: a content-independent motif-based approach. *Online Social Networks and Media*, 3:22–31.

Conover, M., Ratkiewicz, J., Francisco, M., Gonçalves, B., Menczer, F., and Flammini, A. (2011). Political polarization on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5.

Darwish, K., Magdy, W., Rahimi, A., Baldwin, T., and Abokhodair, N. (2018). Predicting online islamophobic behavior after #parisattacks. *The Journal of Web Science*, 4(3):34–52.

Darwish, K., Magdy, W., and Zanouda, T. (2017). Improved stance prediction in a user similarity feature space. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, page 145–148, New York, NY, USA. Association for Computing Machinery.

Darwish, K., Stefanov, P., Aupetit, M., and Nakov, P. (2020). Unsupervised user stance detection on twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):141–152.

Dong, R., Sun, Y., Wang, L., Gu, Y., and Zhong, Y. (2017). Weakly-guided user stance prediction via joint modeling of content and social interaction. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 1249–1258, New York, NY, USA. Association for Computing Machinery.

Dori-Hacohen, S. and Allan, J. (2015). Automated controversy detection on the web. In *European Conference on Information Retrieval*, pages 423–434. Springer.

Easley, D. and Kleinberg, J. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge university press.

Ebrahimi, J., Dou, D., and Lowd, D. (2016). A joint sentiment-target-stance model for stance classification in tweets. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, pages 2656–2665.

Elfardy, H. and Diab, M. (2016). CU-GWU perspective at SemEval-2016 task 6: Ideological stance detection in informal text. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 434–439, San Diego, California. Association for Computational Linguistics.

Ferreira, C. H., Murai, F., Silva, A. P., Almeida, J. M., Trevisan, M., Vassio, L., Mellia, M., and Drago, I. (2021). On the dynamics of political discussions on instagram: A network perspective. *Online Social Networks and Media*, 25:100155.

Ferreira, W. and Vlachos, A. (2016). Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, San Diego, California. Association for Computational Linguistics.

Ferreira, W. and Vlachos, A. (2019). Incorporating label dependencies in multilabel stance detection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6350–6354, Hong Kong, China. Association for Computational Linguistics.

Fiorina, M. P., Abrams, S. A., and Pope, J. C. (2008). Polarization in the american public: Misconceptions and misreadings. *The Journal of Politics*.

Garimella, K., De Francisci Morales, G., Gionis, A., and Mathioudakis, M. (2017a). Reducing controversy by connecting opposing views. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 81–90.

Garimella, K. et al. (2018a). Polarization on social media.

Garimella, K., Gionis, A., Parotsidis, N., and Tatti, N. (2017b). Balancing information exposure in social networks. *Advances in neural information processing systems*, 30.

Garimella, K., Morales, G. D. F., Gionis, A., and Mathioudakis, M. (2018b). Quantifying controversy on social media. *ACM Transactions on Social Computing*, 1(1):1–27.

Garimella, K., Smith, T., Weiss, R., and West, R. (2021). Political polarization in online news consumption. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 152–162.

Ghosh, S., Singhania, P., Singh, S., Rudra, K., and Ghosh, S. (2019). Stance detection in web and social media: A comparative study. In Crestani, F., Braschler, M., Savoy, J., Rauber, A., Müller, H., Losada, D. E., Heinatz Bürki, G., Cappellato, L., and Ferro, N., editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 75–87, Cham. Springer International Publishing.

Gillani, N., Yuan, A., Saveski, M., Vosoughi, S., and Roy, D. (2018). Me, my echo chamber, and i: Introspection on social media polarization. *The Web Conference 2018 - Proceedings of the World Wide Web Conference, WWW 2018*, pages 823–831.

Gokcekus, S., Firth, J. A., Regan, C., and Sheldon, B. C. (2021). Recognising the key role of individual recognition in social networks. *Trends in Ecology & Evolution*, 36(11):1024–1035.

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21.

Guerra, P., Meira Jr, W., Cardie, C., and Kleinberg, R. (2013). A measure of polarization on social media networks based on community boundaries. In *Proceedings of the International AAAI Conference on Web and Social Media*.

Guimaraes, A. and Weikum, G. (2021). X-posts explained: Analyzing and predicting controversial contributions in thematically diverse reddit forums. In *ICWSM*, pages 163–172.

Hamidian, S. and Diab, M. T. (2019). Rumor detection and classification for twitter data. *CoRR*, abs/1912.08926.

Hercig, T., Krejzl, P., Hourová, B., Steinberger, J., and Lenc, L. (2017). Detecting stance in czech news commentaries. *ITAT*, 176:180.

Horawalavithana, S., Ng, K. W., and Iamnitchi, A. (2021). Drivers of polarized discussions on twitter during venezuela political crisis. In *13th ACM Web Science Conference 2021*, pages 205–214.

Iyengar, S., Sood, G., and Lelkes, Y. (2012). Affect, not ideologya social identity perspective on polarization. *Public opinion quarterly*, 76(3):405–431.

Jang, M. and Allan, J. (2018). Explaining controversy on social media via stance summarization. *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*, pages 1221–1224.

Jang, M., Foley, J., Dori-Hacohen, S., and Allan, J. (2016). Probabilistic approaches to controversy detection. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 2069–2072.

Kawintiranon, K. and Singh, L. (2021). Knowledge enhanced masked language model for stance detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4725–4735, Online. Association for Computational Linguistics.

Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information fusion*, 14(1):28–44.

Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing & Its Applications*, 13(3).

Klayman, J. (1995). Varieties of confirmation bias. *Psychology of learning and motivation*, 32:385–418.

Klenner, M., Amsler, M., and Hollenstein, N. (2014). Verb polarity frames: a new resource and its application in target-specific polarity classification. In *KONVENS*, pages 106–115.

Kochkina, E., Liakata, M., and Augenstein, I. (2017). Turing at SemEval-2017 task 8: Sequential approach to rumour stance classification with branch-LSTM. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 475–480, Vancouver, Canada. Association for Computational Linguistics.

Kubin, E. and von Sikorski, C. (2021). The role of (social) media in political polarization: a systematic review. *Annals of the International Communication Association*, 45(3):188–206.

Küçük, D. and Can, F. (2020). Stance detection: A survey. *ACM Comput. Surv.*, 53(1).

Kucher, K., Paradis, C., and Kerren, A. (2018). Visual analysis of sentiment and stance in social media texts. In *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Posters*, EuroVis '18, page 49–51, Goslar, DEU. Eurographics Association.

Kulshrestha, J., Eslami, M., Messias, J., Zafar, M. B., Ghosh, S., Gummadi, K. P., and Karahalios, K. (2017). Quantifying search bias: Investigating sources of bias for political searches in social media. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 417–432.

Lai, M., Farías, D. I. H., Patti, V., and Rosso, P. (2017). Friends and enemies of clinton and trump: Using context for detecting stance in political tweets. *CoRR*, abs/1702.08021.

Lelkes, Y. (2016). Mass polarization: Manifestations and measurements. *Public Opinion Quarterly*, 80(S1):392–410.

Li, C., Porco, A., and Goldwasser, D. (2018). Structured representation learning for online debate stance prediction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3728–3739, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Li, Y. and Caragea, C. (2019). Multi-task stance detection with sentiment and stance lexicons. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6299–6305, Hong Kong, China. Association for Computational Linguistics.

Lima, L., Reis, J. C., Melo, P., Murai, F., Araujo, L., Vikatos, P., and Benevenuto, F. (2018). Inside the right-leaning echo chambers: Characterizing gab, an unmoderated social system. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 515–522. IEEE.

Liu, B. (2010). Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca*.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Liu, C., Li, W., Demarest, B., Chen, Y., Couture, S., Dakota, D., Haduong, N., Kaufman, N., Lamont, A., Pancholi, M., Steimel, K., and Kübler, S. (2016). IUCL at SemEval-2016 task 6: An ensemble model for stance detection in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 394–400, San Diego, California. Association for Computational Linguistics.

Lu, H., Caverlee, J., and Niu, W. (2015). Biaswatch: A lightweight system for discovering and tracking topic-sensitive opinion bias in social media. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 213–222.

Matakos, A., Terzi, E., and Tsaparas, P. (2017). Measuring and moderating opinion polarization in social networks. *Data Mining and Knowledge Discovery*, 31(5):1480–1505.

McPherson, M., Smith-Lovin, L., and Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444.

Mejova, Y., Zhang, A. X., Diakopoulos, N., and Castillo, C. (2014). Controversy and sentiment in online news. *arXiv preprint arXiv:1409.8152*.

Milroy, L. and Llamas, C. (2013). Social networks. *The handbook of language variation and change*, pages 407–427.

Mitchell, J. C. (1974). Social networks. *Annual review of anthropology*, 3:279–299.

Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., and Cherry, C. (2016). SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Mohammad, S. M., Sobhani, P., and Kiritchenko, S. (2017). Stance and sentiment in tweets. *ACM Trans. Internet Technol.*, 17(3).

Morales, A. J., Borondo, J., Losada, J. C., and Benito, R. M. (2015). Measuring political polarization: Twitter shows the two sides of venezuela. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(3):033114.

Moreira, R. C., Vaz-de Melo, P. O., and Pappa, G. L. (2020). Elite versus mass polarization on the brazilian impeachment proceedings of 2016. *Social Network Analysis and Mining*.

Mutlu, E. C., Oghaz, T., Jasser, J., Tutunculer, E., Rajabi, A., Tayebi, A., Ozmen, O., and Garibay, I. (2020). A stance data set on polarized conversations on twitter about the efficacy of hydroxychloroquine as a treatment for covid-19. *Data in brief*, 33:106401.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104.

Pannucci, C. J. and Wilkins, E. G. (2010). Identifying and avoiding bias in research. *Plastic and reconstructive surgery*, 126(2):619.

Pergola, G., Gui, L., and He, Y. (2020). A disentangled adversarial neural topic model for separating opinions from plots in user reviews. *arXiv preprint arXiv:2010.11384*.

Popescu, A.-M. and Pennacchiotti, M. (2010). Detecting controversial events from twitter. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1873–1876.

Rajendran, G., Chitturi, B., and Poornachandran, P. (2018). Stance-in-depth deep neural approach to stance classification. *Procedia Computer Science*, 132:1646–1653. International Conference on Computational Intelligence and Data Science.

Rashed, A., Kutlu, M., Darwish, K., Elsayed, T., and Bayrak, C. (2020). Embeddings-based clustering for target specific stances: The case of a polarized turkey. *CoRR*, abs/2005.09649.

Rathje, S., Van Bavel, J. J., and Van Der Linden, S. (2021). Out-group animosity drives engagement on social media. *Proceedings of the National Academy of Sciences*, 118(26):e2024292118.

Ravi, K. and Ravi, V. (2015). A survey on opinion mining and sentiment analysis. *Know.-Based Syst.*, 89(C):14–46.

Rettore, P. H., Santos, B. P., Campolina, A. B., Villas, L. A., and Loureiro, A. A. (2016). Towards intra-vehicular sensor data fusion. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 126–131. IEEE.

Ribeiro, F. N., Saha, K., Babaei, M., Henrique, L., Messias, J., Benevenuto, F., Goga, O., Gummadi, K. P., and Redmiles, E. M. (2019). On microtargeting socially divisive ads: A case study of russia-linked ad campaigns on facebook. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 140–149.

Roy, S. and Goldwasser, D. (2020). Weakly supervised learning of nuanced frames for analyzing polarization in news media. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7698–7716, Online. Association for Computational Linguistics.

Sailunaz, K., Dhaliwal, M., Rokne, J., and Alhajj, R. (2018). Emotion detection from text and speech - a survey. *Social Network Analysis and Mining (SNAM), Springer*, 8.

Santos, B. P., Silva, L. A., Celes, C. S., Borges Neto, J. B., Peres, B. S., Vieira, M. A. M., Vieira, L. F. M., Goussevskaia, O. N., and Loureiro, A. A. (2016). Internet das coisas: da teoria à prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuıdos*.

Sen, I., Flöck, F., and Wagner, C. (2020). On the reliability and validity of detecting approval of political actors in tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1413–1426, Online. Association for Computational Linguistics.

Shahrezaye, M., Papakyriakopoulos, O., Serrano, J. C. M., and Hegelich, S. (2019). Measuring the ease of communication in bipartite social endorsement networks: A proxy to study the dynamics of political polarization. *ACM International Conference Proceeding Series*, pages 158–165.

Siddiqua, U. A., Chy, A. N., and Aono, M. (2019). Tweet stance detection using an attention based neural ensemble model. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, Volume 1 (Long and Short Papers), pages 1868–1873, Minneapolis, Minnesota. Association for Computational Linguistics.

Sunstein, C. R. (1999). The law of group polarization. *University of Chicago Law School, John M. Olin Law & Economics Working Paper*.

Sunstein, C. R. (2018). *# Republic: Divided democracy in the age of social media*. Princeton University Press.

Swami, S., Khandelwal, A., Singh, V., Akhtar, S. S., and Shrivastava, M. (2018). An english-hindi code-mixed corpus: Stance annotation and baseline system. *CoRR*, abs/1805.11868.

Tachaiya, J., Irani, A., Esterling, K. M., and Faloutsos, M. (2021). Sentistance: Quantifying the intertwined changes of sentiment and stance in response to an event in online forums. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '21, page 361–368, New York, NY, USA. Association for Computing Machinery.

Thomas, M. and Joy, A. T. (2006). *Elements of information theory*. Wiley-Interscience.

Tokita, C. K., Guess, A. M., and Tarnita, C. E. (2021). Polarized information ecosystems can reorganize social networks via information cascades. *Proceedings of the National Academy of Sciences of the United States of America*, 118(50).

Tsakalidis, A., Aletras, N., Cristea, A. I., and Liakata, M. (2018). Nowcasting the stance of social media users in a sudden vote: The case of the greek referendum. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 367–376, New York, NY, USA. Association for Computing Machinery.

Tsytsarau, M., Palpanas, T., and Denecke, K. (2011). Scalable detection of sentiment-based contradictions. *DiversiWeb, WWW*, 1:9–16.

Tucker, J. A., Guess, A., Barberá, P., Vaccari, C., Siegel, A., Sanovich, S., Stukal, D., and Nyhan, B. (2018). Social media, political polarization, and political disinformation: A review of the scientific literature. *Political polarization, and political disinformation: a review of the scientific literature (March 19, 2018)*.

Valensise, C. M., Cinelli, M., and Quattrociocchi, W. (2022). The dynamics of online polarization. *arXiv preprint arXiv:2205.15958*.

Vicario, M. D., Quattrociocchi, W., Scala, A., and Zollo, F. (2019). Polarization and fake news: Early warning of potential misinformation targets. *ACM Transactions on the Web (TWEB)*, 13(2):1–22.

Waller, I. and Anderson, A. (2021). Quantifying social organization and political polarization in online platforms. *Nature*, 600(7888):264–268.

Watts, D. J., Rothschild, D. M., and Mobius, M. (2021). Measuring the news and its impact on democracy. *Proceedings of the National Academy of Sciences*, 118(15).

Weld, G., Glenski, M., and Althoff, T. (2021). Political bias and factualness in news sharing across more than 100,000 online communities. *arXiv preprint arXiv:2102.08537*.

Woodrum, E. and Davison, B. L. (1992). Reexamination of religious influences on abortion attitudes. *Review of religious research*, pages 229–243.

Yang, M., Wen, X., Lin, Y.-R., and Deng, L. (2017). Quantifying content polarization on twitter. In *2017 IEEE 3rd international conference on collaboration and internet computing (CIC)*, pages 299–308. IEEE.

Zhang, B., Yang, M., Li, X., Ye, Y., Xu, X., and Dai, K. (2020). Enhancing cross-target stance detection with transferable semantic-emotion knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3188–3197, Online. Association for Computational Linguistics.

Capítulo 4

Geração de Séries Temporais Utilizando Redes Generativas Adversárias: da Teoria à Prática

Iran F. Ribeiro[1], Breno Krohling[1], Giovanni Comarela[1], Vinícius F. S. Mota[1]

[1]Departamento de Informática - Universidade Federal do Espírito Santo

{iran.ribeiro@edu.ufes, breno.krohling}@edu.ufes.br

{gc.inf, vinicius.mota}@inf.ufes.br

Abstract

Time series is a concept used to model a sequence of data gathered over time. These temporal data usually present properties that allow modeling and forecasting of their applications' scenarios, such as eHealth, intelligent transportation, media recommendation, or any business intelligence application. However, due to business roles and privacy issues, most of the data gathered by enterprises are private and protected. An alternative to tackle these problems is providing the data model instead of the raw data, leading to more transparency and participation of the research community. In such a case, a good data model can provide synthetic data based on the original data, preserving its characteristics while maintaining its privacy. Thus, this chapter discusses the main concepts of modeling time series data. We discuss the main concepts, classical statistical techniques, and deep learning approaches to model the data. In particular, we focus on the generative adversarial network, used initially to transform images, as a novel technique to reproduce time series. We follow with a hands-on lab to model and reproduce synthetic data stemmed from the raw data.

Resumo

Série temporal é um conceito usado para modelar uma sequência de dados coletados ao longo do tempo. Esses dados temporais geralmente apresentam propriedades que

permitem a modelagem e previsão dos cenários de suas aplicações, como eHealth, transporte inteligente, recomendação de mídia, entre outras. No entanto, devido às regras de negócio e problemas de privacidade, a maioria dos dados coletados por empresas é protegida. Uma alternativa é fornecer o modelo de dados ao invés dos dados brutos, levando a mais transparência e participação da comunidade de pesquisa. Nesse caso, um bom modelo de dados pode fornecer dados sintéticos com base nos dados originais, preservando suas características e mantendo sua privacidade. Deste modo, este capítulo discute os principais conceitos para a modelagem de dados de séries temporais. Para isto, são apresentados os principais conceitos, técnicas estatísticas clássicas e abordagens de aprendizado profundo para modelar os dados, em particular, as redes adversárias generativas. Esta última, usada inicialmente para transformar imagens, e agora sendo utilizada como uma nova técnica para reproduzir séries temporais. Por fim, o capítulo apresenta um laboratório prático com implementações para modelar e reproduzir dados sintéticos derivados dos dados brutos.

4.1. Introdução

O aumento de dispositivos de sensoriamento coletando dados continuamente, bem como o armazenamento estruturado de uma enorme quantidade de informações permite uma melhor compreensão da sociedade, natureza, saúde e inteligência de negócios. Em especial, eventos cuja coleta de dados deve ocorrer a cada período podem ser modelados como séries temporais. Exemplos de tais eventos vão desde tráfego de veículos por hora nas ruas e avenidas da cidade, observação da concentração de monóxido de carbono (*CO*) em uma região durante um período, medição de eletrocardiograma (ECG) e monitoramento de acessos simultâneos (carga) em servidores.

A modelagem destes dados como séries temporais permite realizar classificações e predições de informação. Por exemplo, detecção de valores atípicos (*outliers*), agrupamento por semelhança, e predição de eventos, o que agrega de fato valor aos dados brutos. Contudo, embora haja um esforço contínuo para tornar os dados em domínio público para pesquisadores, tais dados podem conter informações restritas, tais como posicionamento ao longo do tempo, transições entre estações-base de uma rede sem fio (ZHENG et al., 2009; Malandrino; Chiasserini; Kirkpatrick, 2018), transições entre sessões de conferências técnicas (SCOTT et al., 2009; RIBEIRO et al., 2021a). Além disso, na prática, dados reais são propensos a conterem erros, a base de dados brutos pode ser grande demais para compartilhamento, falta de dados e exigem garantias de privacidade antes de serem disponibilizados ao público.

Uma abordagem para contornar os problemas acima é a geração de modelos de séries temporais capazes de gerar dados com as mesmas propriedades estatísticas dos dados brutos. Além disso, um modelo generativo permite maior repetibilidade para pesquisas e experimentos, uma vez que cada base gerada é diferente. Deste modo, um

grupo de pesquisa que possua dados sensíveis pode compartilhar com a comunidade científica apenas o modelo gerador da série temporal ao invés dos dados brutos, preservando, assim, a privacidade dos dados. É importante ressaltar que um modelo generativo deve considerar as características intrínsecas de cada cenário e aplicação. Tal heterogeneidade de cenários dificulta uma solução única e que capture as propriedades da aplicação que se deseja gerar dados.

A Figura 4.1 ilustra três cenários de aplicações distintas que geram dados continuamente, tais como (a) mobilidade de tráfego de veículos, monitoramento de redes e dados de sensores de saúde. No lugar de compartilhar os dados brutos, os dados podem ser processados visando identificar o melhor modelo generativo (b). Deste modo, um modelo generativo da série temporal pode ser compartilhado para que demais pesquisadores gerem os modelos sintéticos. Portanto, as primeiras perguntas que surgem e guiam este capítulo são: (1) Como gerar modelos generativos a partir de dados brutos? (2) Como mensurar a qualidade dos modelos geradores?

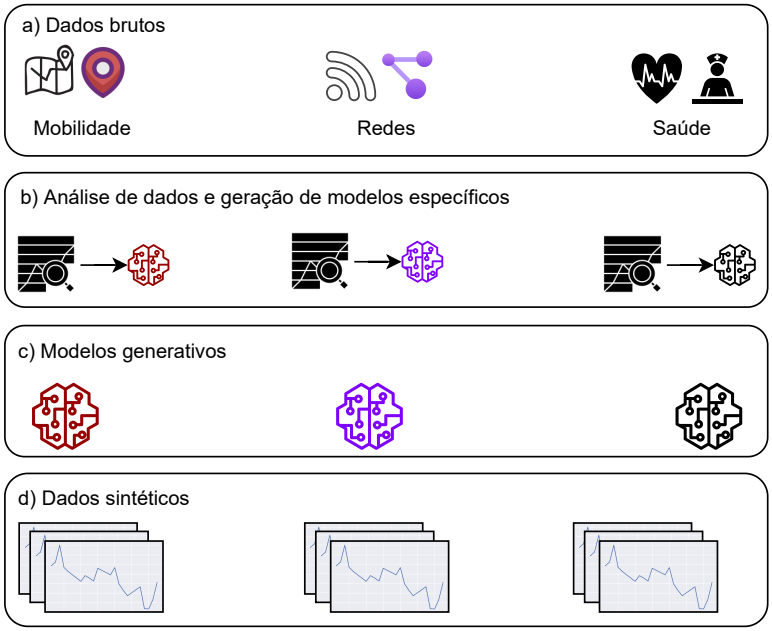


Figura 4.1: Aplicações heterogêneas que geram dados continuamente podem ter modelos capazes de gerar dados sintéticos similares.

Neste sentido, *o foco deste capítulo é apresentar as técnicas disponíveis na literatura para a geração de séries temporais*. Para responder a primeira pergunta, iremos apresentar as técnicas estatísticas clássicas, seguido por uma técnica de aprendizado profundo chamada Redes Generativas Adversárias (*Generative Adversarial Networks* – GANs). As GANs visam a otimização de modelos generativos baseados em aprendizado

profundo, originalmente focadas em projeção e recuperação de imagens (GOODFELLOW et al., 2014). O bom desempenho no campo de visão computacional levou pesquisadores de diversas áreas perceberam o potencial das GANs em gerar dados sintéticos com as características dos dados brutos reais (YI; WALIA; BABYN, 2019).

Adicionalmente, este capítulo discute o desafio de *como mensurar a qualidade do modelo para gerar conjuntos de dados sintéticos*. Normalmente, os modelos em séries temporais são utilizados para fazer previsões. Tais modelos preditivos consideram métricas, como erro absoluto médio (do inglês *Mean Average Error* MAE) ou seus equivalentes, que mensuram os erros de previsão. Tais modelos esperam valores de erros baixos, e um erro igual a zero significa que o modelo é ótimo, isto é, o modelo previu exatamente o valor esperado. No entanto, ao avaliar modelos generativos para reproduzir dados, baixos valores de erros ao comparar os dados sintéticos com os dados reais indicam que os dados sintéticos são praticamente uma cópia do original. Nestes casos, os modelos e os dados sintéticos podem representar um problema, principalmente quando se trata de questões de privacidade. Assim, os modelos generativos requerem métricas que capturem a variabilidade do conjunto de dados sintéticos reproduzidos pelo modelo.

Após a discussão teórica dos conceitos básico, técnicas e métricas, este capítulo apresenta um *hands on* visando a implementação prática para a geração de séries temporais de aluguel de bicicletas em estações de autosserviço nos EUA. Após a geração do modelo, os dados gerados por este modelo serão avaliados de forma qualitativa e quantitativa em relação aos dados originais.

4.2. Motivação e Casos de uso de Geração de Séries Temporais

Os algoritmos de aprendizado profundo têm se mostrado eficientes em diversas áreas de estudo e sendo aplicados para a resolução dos mais diversos tipos de problemas. Nesse sentido, apesar de ser uma tecnologia relativamente recente, ao longo deste capítulo apresentamos brevemente diversos exemplos de aplicação de GANs em séries temporais. Nota-se com os exemplos que, apesar dos desafios e limitações, as GANs podem ser aplicadas com eficiência para problemas envolvendo séries temporais.

4.2.1. Música

A geração dos mais diversos tipos de arte têm se mostrado uma das mais fascinantes aplicações da Inteligência Artificial nas últimas décadas. Sem dúvida, as GANs desempenham um papel fundamental nesse contexto, com a geração de imagens, vídeos e texto, por exemplo. O desempenho das GANs nessas áreas motivou também a investigação da geração de músicas a partir de modelos generativos, entre eles, as GANs. A seguir apresentamos apenas dois exemplos em relação à geração de músicas. Recomendamos a leitura dos trabalhos de (DONAHUE; MCAULEY; PUCKETTE, 2018; ENGEL et al.,

2019) que descrevem abordagens distintas sobre o mesmo assunto.

O primeiro trabalho a propor uma arquitetura de GANs para a geração de músicas é o de (MOGREN, 2016), que já foi mencionado em seções anteriores neste capítulo. Essa arquitetura é muito similar à GAN clássica, composta por um Gerador e um Discriminador, onde cada um dos componentes é uma rede neural recorrente, especificamente, LSTMs. O modelo foi treinado com arquivos de música no formato *midi* contendo características como duração, tom, intensidade e tempo desde o início do último tom. Além disso, cada arquivo representa uma música de um compositor clássico e é possível definir quais compositores serão usados para treinamento do modelo.

Em seu artigo, (DONG HAO-WEN E HSIAO; YANG; YANG, 2018) apresentam os principais desafios envolvendo a geração de notação musical, como dependência temporal, uma estrutura hierárquica interna (um único símbolo e um conjunto do mesmo símbolo possuem sentidos diferentes), necessidade de atender expectativas de ouvintes humanos (coerência, ritmo, tensão) e o fato de uma música geralmente ser composta de diferentes instrumentos. No trabalho, os autores consideram que existem três tipos de música: i) improvisada, onde os músicos executam os instrumentos sem nenhum arranjo pré-definido; ii) composta, em que a música é composta antes da execução; e iii) híbrido, onde os dois tipos são mesclados. Como são tarefas distintas, os autores propõem uma arquitetura específica para cada uma delas.

4.2.2. Mobilidade e Redes

Como mencionamos anteriormente, bases de dados de diversas áreas, como redes e mobilidade, têm seu acesso e divulgação limitados devido a diversos fatores, entre eles, a privacidade. Nesse sentido, ressalta-se que a área de redes é diretamente afetada pela mobilidade das pessoas, pois é tal mobilidade que dita planejamentos ou modificações de estruturas de rede (antenas, roteadores, etc.), bem como o estudo e criação de novas tecnologias. Além disso, a mobilidade também é importante para o estudo de redes sociais de pessoas, por exemplo, entender as características de grupos de indivíduos em uma cidade ou para avaliação de algoritmos de redes oportunísticas. Devido à falta de cenários reais, é comum que novas tecnologias sejam testadas em cenários simulados, que podem falhar em representar o ambiente em que a tecnologia será de fato utilizada. Nesse contexto, as GANs têm se mostrado uma alternativa a permitir que mais bases de dados representando cenários reais de mobilidade e, consequentemente, de redes, sejam disponibilizadas.

Exemplos de como a geração de mobilidade é feita com o uso de GANs podem ser vistos nos trabalhos de (AMIRIAN; HAYET; PETTRÉ, 2019; RAO et al., 2020), em que os autores criam modelos para geração de trajetórias feitas por pessoas e veículos ou em (ZHANG et al., 2019b, 2019a) em que são gerados dados representando o tráfego de carros em ruas de cidades. A partir de dados de redes de celulares (HUANG; KUR-

NIAWAN; SUN, 2022) propõem a utilização de GANs como uma forma de identificar anomalias em métricas importantes para o gerenciamento de redes de celular. Com o objetivo de aumentar a eficiência de algoritmos de previsão em dados de redes de celular, (WANG et al., 2020) propõem o uso de GANs como técnica de *data augmentation* (técnica onde mais amostras são adicionadas à base de dados).

Outros cenários interessantes abordando a mobilidade podem ser vistos nos trabalhos de (CHEN et al., 2019; ZHANG, 2019), que tratam da geração de dados para predição da formação de conexões entre pessoas, por exemplo, por *bluetooth* ou redes *ad hoc*. Esse tipo de dado pode ser usado em experimentos considerando diversos cenários, por exemplo, para comparar novos algoritmos com outros já conhecidos ou para avaliar algoritmos em novas tecnologias, como o 5G.

4.2.3. Saúde

Na área da saúde, uma das grandes aplicações de séries temporais é poder representar a evolução de pacientes ao longo do tempo. Essa representação é importante, por exemplo, para realização de previsões sobre a condição do paciente a partir da evolução do seu quadro clínico. Da mesma forma, esse tipo de dado pode auxiliar a entender como se dá a ocupação de leitos de hospitais ao longo do tempo, considerando que cada enfermidade exige tempos diferentes de tratamento e recuperação.

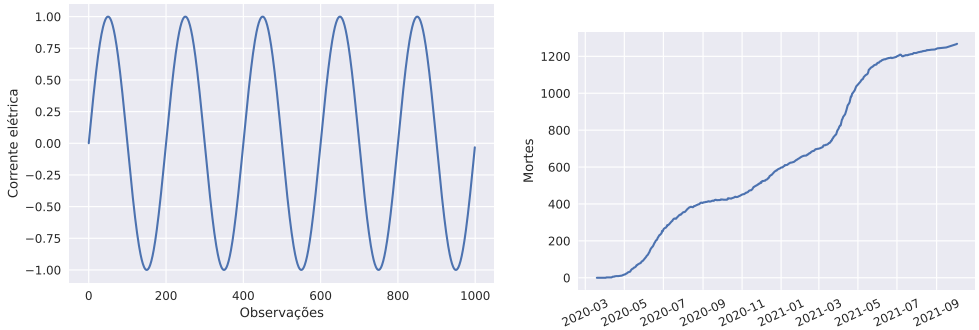
Um dos primeiros trabalhos a utilizar GANs para geração de dados médicos foi o de (ESTEBAN; HYLAND; RÄTSCH, 2017). Especificamente, os autores utilizam séries temporais de uma UTI para treinamento do modelo, considerando que tais séries podem conter dados numéricos ou categóricos. Uma aplicação direta dos dados gerados pelo modelo, é o treinamento de médicos que, segundo o autor, é geralmente feito por dados simulados que podem não representar a realidade de uma UTI. Em outro trabalho, (KADRI et al., 2022) utilizam dados de hospitais para prever o tempo em que determinados pacientes ficam nas salas de emergência. Essa é uma informação importante para diretores de hospitais, pois é possível prever e otimizar, com certa segurança, como os atendimentos no hospital ocorrem.

Nesse sentido, dados faltantes nas bases de dados são um grande problema em diversas áreas. Por exemplo, para realizar tarefas de previsões é esperado uma quantidade predefinida de variáveis de entrada. Nesse contexto, diversas técnicas podem ser utilizadas para tratar desses dados faltantes, entretanto, podem não ser ideais considerando a questão temporal dos dados. Como exemplos de trabalhos que utilizam GANs para tratar desse problema tem-se (LUO et al., 2018) e (OH et al., 2021).

4.3. Introdução às Séries temporais

Uma série temporal é um conjunto de observações registradas em um tempo específico (BROCKWELL; DAVIS, 2009). Cada observação pode representar, simultaneamente, uma ou mais variáveis. Deste modo, as séries temporais podem ser aplicadas em diversas áreas: no audiovisual, pode modelar dados de música (e.g, músicas)(ENGEL et al., 2019); em mobilidade humana, auxilia na análise e modelagem de trajetórias feitas por pessoas (RAO et al., 2020); na área de redes móveis, pode ser aplicada na modelagem de contatos feitos entre pessoas através de *bluetooth* (ZHANG, 2019); e na medicina, pode ser utilizada para análise e modelagem do tempo que um paciente demora em uma sala de emergência (KADRI et al., 2022).

Uma série temporal pode ser categorizada em dois tipos, discreta ou contínua, conforme os intervalos de tempo entre cada observação. A série temporal será discreta quando o intervalo de observações (t) é um conjunto discreto e cada observação ocorre em intervalos de tempo fixo, por exemplo, à cada segundo, minuto, ou hora. Por outro lado, a série será contínua quando as observações são registradas continuamente em um determinado intervalo de tempo, por exemplo, durante 1 minuto, são registradas todas a observações que ocorrem. Como exemplo, a Figura 4.2a mostra uma série temporal de uma corrente que passa por um resistor (contínua) durante 1 segundo e a Figura 4.2b representa as mortes (acumuladas) de Covid-19 entre março de 2020 e setembro de 2021, em Vitória-ES (discreta). Neste sentido, o foco principal deste capítulo será nas séries temporais discretas.



(a) Corrente passando por um resistor (b) Mortes causadas por Covid19

Figura 4.2: Exemplos de séries temporais: contínua, com a corrente que passa por um resistor (a); e discretas, com as mortes causadas por Covid-19 em Vitória-ES (b).

Uma série temporal pode possuir quatro propriedades que auxiliam no seu entendimento. Como estão diretamente relacionadas à situação real que gera a série temporal, seja por efeito natural, humano ou ambos, essas propriedades podem informar também

possíveis causas e correlações entre dados distintos. As quatro propriedades são descritas a seguir (CHATFIELD, 2003):

- **Sazonalidade**: são variações nas observações dos dados que ocorrem em um certo período de tempo (ou temporadas). Uma série temporal contendo a temperatura de um país é um exemplo de série temporal com sazonalidade, com variação anual (o inverno tende a ser mais frio e verão mais quente).
- **Ciclos**: é um tipo de variação similar à sazonalidade, sendo geralmente causada por processos externos à natureza da série temporal. Diferente da sazonalidade, os ciclos podem ocorrer por tempos variados e em intervalos de tempo distintos. As variações de temperatura durante um dia são um exemplo de variação cíclica.
- **Tendência**: a tendência pode ser definida como uma mudança na média da série temporal ocorrendo por um longo período. Essa variação pode fazer com que a média da série temporal sofra um aumento (tendência positiva), uma diminuição (tendência negativa) ou nula (tendência horizontal).
- **Irregularidades**: são alterações aleatórias nos dados, sem nenhum padrão específico. Geralmente são variações de curta duração.

Uma série temporal pode ser entendida como uma sequência de observações, que assim como muitos outros processos reais, são aleatórias, e podem possuir propriedades probabilísticas. Dessa forma, uma das formas mais comuns de se entender e modelar uma série temporal é tentar identificar o processo estocástico que a pode ter gerado.

Segundo (SHUMWAY; STOFFER, 2000), um processo estocástico é um conjunto de variáveis aleatórias $\{X_t\}$ onde t, é um valor discreto que varia no intervalo dos números inteiros ($t = 0, \pm 1, \pm 2, ...$). Nesse sentido, uma forma de se descrever um processo estocástico é entender os momentos do processo, nesse caso, o primeiro e segundo momentos, os quais são a média (primeiro momento) e variância e autocovariância (segundo momento) (CHATFIELD, 2003). A autocovariância (γ_k), nesse caso, basicamente calcula a covariância entre valores atuais (t) e futuros ($t+k$) de um processo:

$$\gamma_k = \frac{\sum_{t=1}^{N-k}(X_t - \mu)(X_{t+k} - \mu)}{N-k} \tag{1}$$

onde k representa um intervalo de defasagem entre as observações, μ é a média e N é o número de observações do processo.

Uma das classes mais importantes de processos estocásticos são os processos estacionários. Há diferentes tipos de níveis de restrições para que um processo seja

considerado estacionário. Uma série temporal será estritamente estacionária se as distribuições conjuntas de $X_{t_1},...,X_{t_n}$ e $X_{t_1+\tau},...,X_{t_n+\tau}$ são iguais, ou seja, as distribuições não dependem do tempo t ou de qualquer deslocamento τ aplicado à série temporal. Essa definição, entretanto, possui um nível de restrição muito alto, sendo comum a utilização de uma definição com menos restrições, conhecida como estacionária de segunda ordem. Logo, uma série temporal será estacionária quando sua média for constante e sua função de autocovariância dependa somente da diferença entre intervalos distintos ($t_2 - t_1$).

A análise estatística das propriedades de uma série temporal permite a identificação da categoria de processo estocástico da qual ela faz parte. Assim, alguns dos processos mais comuns que auxiliam na classificação e modelagem de séries temporais são (CHATFIELD, 2003):

- **Puramente aleatório**: um processo que consiste em uma sequência de variáveis aleatórias $\{Z_t\}$ que são mutuamente independentes e identicamente distribuídas. Esse tipo de processo possui média e variância constantes e, como sua função de autocovariância não depende do tempo, será estacionário de segunda-ordem.

- **Passeio Aleatório**: supondo-se que $\{Z_t\}$ seja um processo puramente aleatório com média μ e variância σ_z^2. Um processo $\{X_t\}$ será um passeio aleatório se $X_t = X_{t-1} + Z_t$. Como a média e a variância são dependentes do tempo, o processo é não-estacionário.

- **Média móveis**: Considerando que $\{Z_t\}$ seja um processo aleatório com média 0 e variância σ_z^2, um processo $\{X_t\}$ será de médias móveis de ordem q, representado por MA(q) (do inglês, *Moving Average*), se:

  $$X_t = \beta_0 Z_t + \beta_1 Z_{t-1} + ... + \beta_q Z_{t-q}.$$

  onde β_i, com $i = 0,...,q$, são constantes. Esse processo é estacionário, pois sua função de autocovariância não depende do tempo t e a média é constante.

- **Auto-regressivo**: Sendo $\{Z_t\}$ um processo puramente aleatório com média 0 e variância σ_z^2, um processo $\{X_t\}$ será auto-regressivo de ordem p se:

  $$X_t = \alpha_1 X_{t-1} + ... + \alpha_p X_{t-p} + Z_t. \quad (2)$$

  Considerando o caso mais simples, onde o processo é de primeira ordem (p=1), tem-se que a Equação 2 pode ser simplificada para:

  $$X_t = \alpha_1 X_{t-1} + Z_t. \quad (3)$$

nesse caso, considerando que $-1 \leq \alpha \leq +1$, podemos rescrever a Equação 3 como um processo MA de ordem infinita:

$$X_t = Z_t + \alpha Z_{t-1} + \alpha^2 Z_{t-2} + \ldots$$

Dessa forma, a função de autocovariância para esse tipo de processo não depende do tempo t e o processo será estacionário (de segunda-ordem) quando $-1 \leq \alpha \leq +1$. Para o caso geral ($p > 1$), referenciamos o livro de Chatfield (2003). Assim como em um processo de médias móveis, um processo auto-regressivo é normalmente abreviado para AR(p) (*Auto-Regressive*, na sigla em Inglês).

Considerando os tipos de processos apresentados, é comum a definição de métodos capazes de modelar mais de um processo simultaneamente. Como foi visto, um processo Puramente Aleatório é usado na definição dos processos de Passeio Aleatório, Média Móvel e Auto-regressivos. Além disso, um dos tipos de modelos mais úteis para séries temporais é resultado da combinação de um processo Auto-regressivo (AR) e um processo de Médias Móveis (MA), resultando em um modelo conhecido como ARMA. Desta forma, um processo ARMA de ordem i(p,q), abreviado para ARMA(p,q), é definido por:

$$X_t = \alpha_1 X_{t-1} + ... + \alpha_p X_{t-p} + Z_t + \beta_1 Z_{t-1} + ... + \beta_q Z_{t-q}.$$

A vantagem da utilização deste tipo de modelo é que mais comum que uma série temporal estacionária seja descrita por um processo ARMA contendo poucos parâmetros do que por um processo MA ou AR sozinhos (CHATFIELD, 2003). Nesse sentido, na prática, a maioria das séries temporais não são estacionárias, dificultando a utilização de modelos ARMA. A partir dessas limitações, surgiu o modelo ARIMA, que será descrito com mais detalhes a seguir.

4.3.1. ARIMA

O ARIMA (*Auto-Regressive Integrated Moving Average*) é um modelo estatístico linear bastante conhecido na análise de séries temporais (ZHANG, 2003). Ele é uma generalização do modelo ARMA, sendo utilizado em séries temporais não-estacionárias, ou seja, uma série cujos dados oscilam sobre uma média que pode variar temporalmente.

Um modelo ARIMA é definido utilizando-se os parâmetros p, d e q, onde p representa o número de parâmetros auto-regressivos, d o número de diferenciações para tornar a série estacionária e q o número de parâmetros utilizado nas médias móveis. Um modelo ARIMA definido pelos parâmetros p, d e q é comumente representado por ARIMA(p, d, q).

Existem diversas metodologias que podem ser utilizadas para se encontrar os parâmetros p, d e q de um modelo ARIMA, sendo a de Box & Jenkins (BOX et al., 2015) a mais conhecida delas. Nessa metodologia, existem 3 etapas fundamentais (Identificação do modelo, Estimação dos parâmetros e Verificação do modelo), descritas a seguir.

4.3.1.1. Identificação do processo

Nesta fase, identifica-se o processo estocástico que gerou os dados. Para isso, algumas técnicas são utilizadas para se encontrar a estrutura do modelo, ou seja, os parâmetros p, d e q. O parâmetro d pode ser identificada por meio de uma sequência de diferenciações aplicadas à série temporal para torná-la estacionária. O número de vezes que a diferenciação precisa ser realizada define o valor de d.

Os parâmetros p e q podem ser identificados ao se analisar, respectivamente, as funções de autocorrelação parcial e de autocorrelação (respectivamente, PACF e ACF, nas siglas em inglês) utilizando-se a série que já passou pelo processo de diferenciação para se tornar estacionária. Cada uma dessas funções exibe a correlação entre a série temporal e um certo número de defasagens. Assim, elas representam como os coeficientes de correlação se comportam em diferentes números de defasagens.

Na Figura 4.3 tem-se uma série temporal, gerada por um processo ARIMA$(1,0,0)$ que representa a variação do PIB de um país entre 1900 e 2000. Nota-se que a série varia entre -1.5 e 2.5, mas não há tendências de crescimento ou decrescimento ao longo tempo, ou seja, a média não varia em função do tempo, logo, $d = 0$. Na Figura 4.4 têm-se os gráficos PACF e ACF da série temporal. Pode-se observar na Figura 4.4a que, após o número de defasagem 1, os coeficientes são muito próximos de zero 0, indicando que a série pode ser representada por um processo com $p = 1$. Além disso, a Figura 4.4b mostra que os valores dos coeficientes decrescem à medida que se aumenta o número de defasagens, indicando ser improvável que a série seja gerada por um processo de médias móveis, ou seja, $q = 0$.

4.3.1.2. Estimação dos Parâmetros

Nesta etapa deseja-se estimar os valores de β_i e α_i das Equações 4.3 e 2 considerando os valores de p e q encontrados na etapa anterior. Diversos procedimentos podem ser utilizados para isso, como funções de verossimilhança e de somas dos quadrados, estimação não-linear, análise dos processos específicos (caso seja verificado que a série é gerada por um processo auto-regressivo, de médias-móveis ou uma combinação de ambos) e estimação usando o Teorema de Bayes. Uma descrição mais detalhada de tais procedimentos pode ser encontrada em Box et al. (2015).

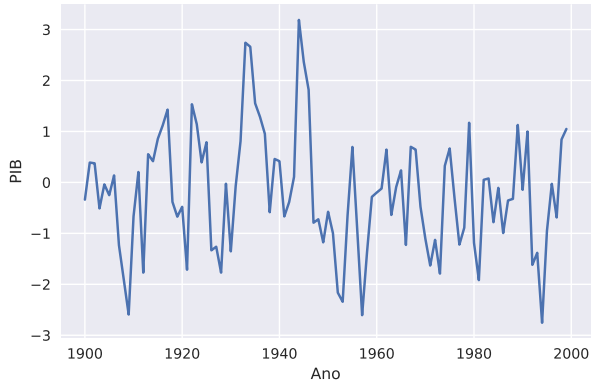
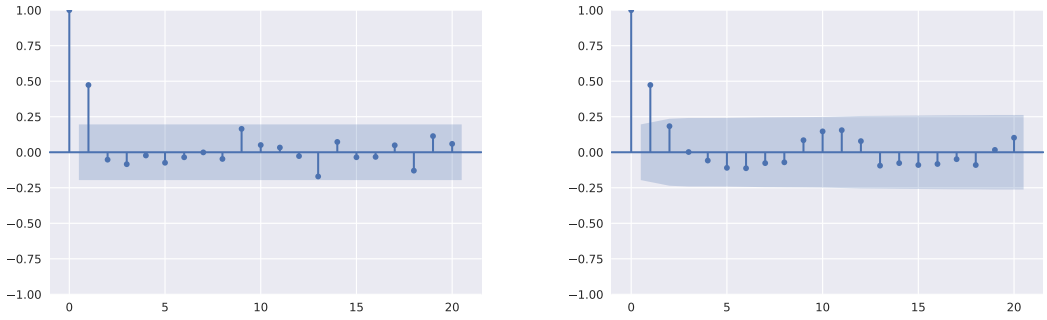


Figura 4.3: Exemplo de série temporal estacionária.



(a) PACF (b) ACF

Figura 4.4: Gráficos PACF (a) e ACF (b) de um processo ARIMA$(1,0,0)$.

4.3.1.3. Verificação

Esta etapa consiste na realização de testes estatísticos para avaliar o desempenho do modelo ARIMA ajustado à série temporal. A ideia geral é identificar se o modelo definido está adequado para a série em questão e quais modificações no modelo precisam ser realizadas para garantir uma melhor adequação.

Um dos testes conhecidos é baseado na análise dos resíduos. A ideia é verificar se os resíduos do modelo ajustado apresentam algum tipo de dependência temporal, o que pode ser feito através dos gráficos de autocorrelação parcial anteriormente citados. Caso seja identificada qualquer tipo de dependência, o modelo não conseguiu capturar corretamente as dependências temporais da série original.

A metodologia de Box & Jenkins é largamente utilizada para definição e escolha

dos parâmetros de um modelo ARIMA. Entretanto, existem alternativas mais rápidas e livres de má interpretação das características da série temporal, que usam a metodologia de Box & Jenkins internamente. Por exemplo, em alguns casos, não é simples identificar se uma série é estacionária e, caso seja, quantas vezes a diferenciação precisa ser feita para torná-la estacionária. Dessa forma, algumas linguagens de programação, como o Python e R, possuem bibliotecas que facilitam as análises preliminares e possibilitam a identificação automática dos parâmetros do modelo.

É importante ressaltar, contudo, que o ARIMA tende a ser mais eficiente em bases de dados pequenas (MONTGOMERY; HINES, 1980). Ainda assim, pela facilidade em se encontrar os parâmetros p, d e q e por ser um modelo relativamente rápido de se treinar, o ARIMA mostra-se um bom modelo para se realizar as primeiras análises dos dados, indicando se um modelo não linear, por exemplo, uma Rede Neural Recorrente, precisa ser ou não utilizado, evitando o desperdício de tempo e de poder computacional. Embora o ARIMA seja comumente utilizado para a previsão de novas observações de séries temporais, também pode ser aplicado em geração de dados sintéticos. Para isso, ajusta-se um modelo a uma série temporal e simula-se uma nova série com base nos parâmetros do modelo que melhor se ajustam à série.

4.4. Análises de séries temporais

De modo geral, as séries temporais são utilizadas para três tipos de análises: Classificação, predição e modelagem da curva. Sendo esta última, a que permite entender a série para que realizar a geração de dados.

4.4.1. Predição

No problema da predição de séries temporais, tem-se um ou mais conjuntos de dados sequenciais e deseja-se realizar algum tipo de previsão com base nas informações obtidas dos dados. Nesse sentido, a previsão de séries temporais pode ser dividida em três métodos (CHATFIELD, 2003):

- **Subjetivo**: utiliza julgamentos subjetivos, intuição e qualquer informação relevante sobre os dados.
- **Univariado**: as previsões são feitas com base em valores passados de uma única série temporal
- **Multivariado**: nesse método, a previsão de uma variável depende, parcial ou totalmente, de valores de uma ou mais séries temporais adicionais.

Conhecimentos subjetivos sobre um determinado problema podem aumentar a eficiência dos métodos usados. De fato, na prática, é muito comum uma combinação dos

métodos, por exemplo, em que se faz uma previsão univariada ou multivariada com base em informações subjetivas que se tem das séries temporais (CHATFIELD, 2003).

Como um exemplo de previsão utilizando o método univariado, tem-se um gerente que deseja saber a variação de estoque de um certo produto durante os próximos 3 meses. Dado que ele tenha valores passados suficientes do estoque do produto (2 anos de dados, por exemplo), o gerente pode utilizar esses valores e estimar a variação do estoque ao longo dos meses desejados. Nesse sentido, os valores estimados podem ser obtidos com base no conhecimento subjetivo que o gerente possui do histórico de vendas. Existem dezenas de trabalhos na literatura baseados em previsão de séries univariadas. Nesse sentido, referenciamos o *survey* de (LIM; ZOHREN, 2021), o qual analisa trabalhos que utilizam técnicas baseadas em redes neurais para previsão de séries temporais univariadas.

É muito comum, contudo, que diversos fatores possam influenciar a variação de uma determinada série temporal. Isso pode fazer com que seja necessário que os métodos considerem mais de uma série temporal para realizar previsões. Como exemplo dessa situação, pode-se citar os diversos indicadores econômicos de um país, mensurados mensalmente, como: índices de preços no varejo, porcentagem de desemprego e índice da bolsa de valores. A partir desses indicadores, pode ser de grande interesse a construção de um modelo que descreva as relações presentes entre as variáveis e então utilizá-lo para realizar algum tipo de predição (e.g., variação da inflação). O leitor interessado em saber mais sobre estudos que realizam previsões multivariadas pode consultar o trabalho de (BEERAM; KUCHIBHOTLA, 2021).

4.4.2. Classificação

Diversos autores têm estudado formas de se classificar séries temporais com base em suas características. Diferente da classificação tradicional, entretanto, a ordem dos atributos possui um papel fundamental durante a classificação (BAGNALL et al., 2017). Na prática, o problema de classificação pode ser resumido em, dado um conjunto séries temporais não rotuladas, mapear cada uma delas a uma classe de um conjunto predefinido de classes (WEI; KEOGH, 2006).

Nesse caso, diferentes técnicas podem ser usadas para a realização da classificação, como as técnicas baseadas em características, baseadas em modelos e aprendizado de máquina. No primeiro caso, as características podem ser extraídas de vários domínios: tempo, frequência e ondas da série temporal. Entre as abordagens baseadas em modelos tem-se o ARIMA (vide Seção 4.3.1), modelos Gaussianos e Bayesianos (SYKACEK; ROBERTS, 2001; POVINELLI et al., 2004; KOTSIFAKOS; PAPAPETROU, 2014). Por fim, dentre as abordagens baseadas em aprendizado de máquina, podemos citar as árvores de classificação e Máquina de Vetores de Suporte (DOUZAL-CHOUAKRIA; AMBLARD, 2012; KAMPOURAKI; MANIS; NIKOU, 2008).

O trabalho de (MAHARAJ; ALONSO, 2014) é um exemplo de trabalho que demonstra a classificação de séries temporais sintéticas considerando o domínio temporal, mais especificamente, ondas. No trabalho os autores simularam sinais de eletrocardiogramas variando um parâmetro λ de forma que, quando $\lambda > 1$, o sinal passa a representar uma pessoa com sofreu um ataque cardíaco (*Acute Myocardial Infarction* - AMI, em inglês). Assim, é possível analisar a variação e a correlação das ondas para identificar quando ocorreu um ataque cardíaco, como foi proposto no trabalho. Esse tipo de análise possui aplicações práticas em situações reais, auxiliando na prevenção e tratamento de pessoa com problemas cardíacos.

4.4.3. Geração

O acesso a bases de dados é fundamental em diversas áreas de pesquisa, como agrupamento de dados, classificação, previsão e detecção de anomalias. Na prática, contudo, existem diversos desafios que limitam ou impedem a utilização eficiente de certos tipos de dados, por exemplo:

- **Tamanho da base de dados**: dependendo do objetivo de utilização dos dados, o tamanho (número de registros) pode impactar significativamente os resultados. No entanto, em algumas situações, pode ser inviável a obtenção de mais dados da fonte original. Por exemplo, em medicina, alguns tipos de doenças ocorrem em poucas pessoas no mundo todo. De forma similar, alguns processos industriais podem levar horas para serem concluídos, de forma que é inviável esperar até que uma grande quantidade de dados esteja disponível para ser utilizada.
- **Dados faltantes com erro**: em muitas situações, é comum ocorrerem erros no momento em que os dados são armazenados. Consequentemente, isso faz com que porções dos dados não possam ser usadas.
- **Privacidade**: A preocupação com a privacidade de dados pessoais tem se intensificado nos últimos anos. A criação da LGPD (BRASIL, 2018) no Brasil é resultado direto dessa preocupação. Nesse sentido, apesar de estarmos em uma era em que nossos dados são coletados diariamente por diversas empresas, a divulgação desses dados depende de muitos processos e pode não ocorrer, devido às questões de privacidade. Além disso, mesmo que as próprias empresas tratem da privacidade com algum tipo de método (e.g., anonimização e inserção de ruído), a utilidade dos dados pode ficar comprometida.

Nesse sentido, a geração de dados sintéticos torna-se uma alternativa para minimizar os impactos das limitações mencionadas anteriormente. É importante ressaltar que no contexto deste minicurso, o termo geração é sinônimo do termo simulação, já que o objetivo, de fato, é simular um cenário (conjunto de dados) real. Por exemplo, a partir de

dados sintéticos, conhecendo-se a distribuição dos dados, uma base de dados pequena pode ser aumentada, lacunas dos dados podem ser preenchidas e dados que simulem os dados reais podem ser disponibilizados publicamente. O maior interesse na geração de dados, principalmente nesse último caso, é permitir maior repetibilidade para pesquisas e experimentos, uma vez que cada base gerada é diferente. Apesar dos benefícios, o uso de dados sintéticos é desafiador, pois é necessário garantir que, de fato, os dados gerados representem a realidade do problema que se deseja tratar.

Contudo, como foco deste minicurso, este ainda é um campo de estudo pouco explorado na literatura quando se trata da geração de séries temporais a partir de um conjunto de dados reais. Além dos desafios que é gerar dados com características realistas, a dependência temporal dos dados deve ser considerada (LIN et al., 2020). Por exemplo, suponha que pesquisadores da área de medicina desejem estudar a evolução do quadro clínico de pessoas com uma determinada doença durante um ano. Por questões de privacidade, o hospital pode não divulgar os dados dos pacientes. Uma alternativa, então, é o hospital disponibilizar um modelo, obtido a partir dos dados reais, que gere dados sintéticos para a realização da pesquisa.

Em teoria, desde que os dados possam ser organizados como séries temporais, o mesmo princípio pode ser aplicado para dados de outros tipos de problemas. Na prática, há exemplos de geração de séries temporais de sensores (ALZANTOT; CHAKRABORTY; SRIVASTAVA, 2017), pacientes em hospital (ESTEBAN; HYLAND; RÄTSCH, 2017) e rede elétrica inteligente (ZHANG et al., 2018). Nesse sentido, as técnicas de geração vão desde modelos essencialmente estatísticos ou lineares (SINGH; RAY, 2021) a modelos baseados em aprendizado profundo (YOON; JARRETT; SCHAAR, 2019). Neste minicurso, apresentamos o último caso, os modelos conhecidos como Redes Generativas Adversárias, uma das técnicas mais recentes e mais bem sucedidas em geração de dados.

4.5. Introdução ao Aprendizado de Máquina

Inteligência artificial, aprendizado de máquina e aprendizado profundo são paradigmas que estão diretamente conectados. De modo geral, todos eles buscam representar o comportamento humano ao realizar alguma tarefa. Segundo Chollet (2021) a diferença entre tais conceitos é dada pela forma como cada algoritmo adquiri a base de conhecimento utilizada para realizar tais tarefas.

Algoritmos de inteligência artificial são desenvolvidos de maneira que as regras, ou conhecimento, são pré-definidos ao nível de modelo. Dessa maneira, dado um conjunto de dados e as regras pré-definidas, o algoritmo consegue inferir uma resposta. Ao contrário dos algoritmos de inteligência artificial, algoritmos de aprendizado de máquina baseiam-se na premissa do aprendizado através de exemplos. Nesse paradigma, as regras são aprendidas por padrões presentes nos dados e respostas previamente conhecidos

em um processo de treinamento e, com o conjunto de regras aprendidas, o algoritmo deve conseguir inferir conclusões ao processar novas amostras de dados pertencentes ao mesmo domínio. Por fim, aprendizado profundo é um sub-campo de aprendizado de máquina onde modelos são compostos por camadas de redes neurais artificiais, treinadas usando algoritmos de otimização em um processo iterativo.

A seguir, apresentamos com mais detalhes alguns dos conceitos presentes nos paradigmas citados. Primeiro, introduzimos os tipos de problemas de aprendizado de máquina. Após isso, serão apresentados os conceitos de redes neurais artificiais.

4.5.1. Tipos de Problema de Aprendizado de Máquina

Problemas de aprendizado de máquina podem ser classificados de acordo com as características dos dados a serem utilizados na etapa de treinamento. Dentre eles, os de Aprendizado Supervisionado e Aprendizado Não-Supervisionado englobam a maioria dos problemas existentes. Para ser caracterizado com um problema de Aprendizado Supervisionado, o conjunto de dados utilizado no processo de treinamento do algoritmo deve possuir, além dos dados de entrada, um dado de saída referente a cada amostra de entrada. Desse modo, o algoritmo busca aprender uma função de modo que possa mapear todos os dados de entrada $\mathbf{x}$ em um dado de saída y. Comumente nos referimos a esse tipo de dados como rotulados. Porém, muitos problemas não possuem dados rotulados, seja pela quantidade de dados presentes em um conjunto de dados, o que geraria um esforço elevado para rotular todos as amostras manualmente, ou pela falta de pessoas com conhecimento especializado que possa realizar tal trabalho. Problemas que envolvem tal tipo de dados (não rotulados), são conhecidos como problemas de Aprendizado Não-Supervisionado. De acordo com Goodfellow, Bengio e Courville (2016), algoritmos supervisionados buscam encontrar uma saída y dado uma amostra de entrada $\mathbf{x}$ ao estimar $P(y|\mathbf{x})$ e algoritmos Não-supervisionados buscam encontrar a distribuição de probabilidade $P(\mathbf{x})$ que gerou os dados.

Problemas de Aprendizado Supervisionado e Não-supervisionado ainda podem ser internamente agrupados de acordo com seu objetivo final. Dentre os problemas existentes, os mais comuns são: 1) Classificação, 2) Predição/Regressão e 3) Agrupamento. A seguir introduzimos cada um desses problemas.

4.5.1.1. Classificação

Problemas de classificação buscam encontrar uma função capaz de mapear uma entrada $\mathbf{x}$ em sua respectiva saída y, onde y representa um valor categórico, ou rótulo, relacionado a entrada do problema (GOODFELLOW; BENGIO; COURVILLE, 2016).

De modo prático, suponha que queremos classificar se um e-mail recebido é um

spam. Inicialmente, necessitamos de uma base de dados de treinamento $X = \{\mathbf{x}_0, \cdots, \mathbf{x}_N\}$, e seus respectivos rótulos $Y = \{y_0, \cdots, y_N\}$ informado se o e-mail em questão é considerado spam. O algoritmo deve então, de maneira supervisionada, aprender uma função capaz de mapear as amostras $\mathbf{x} \in X$ da base de dados de treinamento ao seu respectivo rótulo y. Posteriormente, a função aprendida deve ser capaz de realizar a mesma tarefa para amostras além do grupo de treinamento.

4.5.1.2. Regressão

Problemas de regressão possuem como objetivo prever um valor numérico. Ao contrário de problemas de classificação que querem prever um valor categórico pertencente a um conjunto de rótulos, problemas de regressão irão nos fornecer uma saída contida no conjunto de números reais. Dessa forma, em problemas de regressão, um modelo busca aprender uma função do tipo $f : \mathbb{R}^n \rightarrow \mathbb{R}$ (GOODFELLOW; BENGIO; COURVILLE, 2016).

Dado um conjunto n de características, uma regressão linear pode ser descrita pela Equação 4

$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_b x_n + b, \tag{4}$$

onde x_i representa a i-ésima característica, w_i representa seu peso, treinável associado e b representa o *bias*.

A fim de caracterizar tal problema, podemos utilizar como exemplo de regressão a tarefa de estimar o número de pessoas que irão assistir um determinado video na plataforma YouTube, baseando-se na relação das características de vídeos similares com seus respectivos número de visualizações.

4.5.1.3. Agrupamento

Por fim, problemas de agrupamento buscam encontrar protótipos que possam sumarizar os dados de estudo de acordo com seus padrões e estruturas. O objetivo final do algoritmo é de agrupar os dados dentre um número finito de possíveis grupos.

Segundo Wunsch e Xu (2008), um algoritmo de agrupamento busca agrupar o conjunto de dados $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, onde $\mathbf{x}_i \in \mathbb{R}^n$, em k grupos $C = \{C_1, \cdots, \cdots, C_k\}$ com $(k \leq N)$, seguindo as seguintes restrições:

1. $C_i \neq \emptyset$; $i = 1, ..., k$;

2. $\bigcup_{i=1}^{k} C_i = X$;

3. $C_i \cap C_j = \emptyset$; $i, j = 1, ..., k$ e $i \neq j$.

O processo pela busca de partições está relacionado com a escolha de uma métrica de comparação e uma função objetivo a ser otimizada. Por exemplo, o algoritmo *K-means* busca agrupar os dados ao minimizar a variância da partição. Esse processo é feito ao minimizar os erros quadrados entre as amostras o centroide de um grupo C_i.

Algoritmos de agrupamento podem ser separados de acordo com sua abordagem de processamento dos dados. Algoritmos de partição buscam agrupar os dados em um conjunto finito de grupos pré-definidos, sem necessidade de estruturas mais complexas. Algoritmos hierárquicos não possuem um número de partições pré-definidas nas quais os dados devem ser agrupado. Essa abordagem utiliza de uma estrutura de árvore para encontrar o número ideal de partições para cada conjunto de dados. Algoritmos hierárquicos podem apresentar estratégia de agrupamento do tipo *top-down*, onde inicialmente existe um grupo com todos os dados que são recursivamente separados em partições menores, ou do tipo *bottom-up*, onde cada amostra dos dados representa uma partição que são posteriormente reagrupadas de acordo com suas características. Mais detalhes referentes aos tópicos abordados podem ser encontrados em Zaki e Jr (2020).

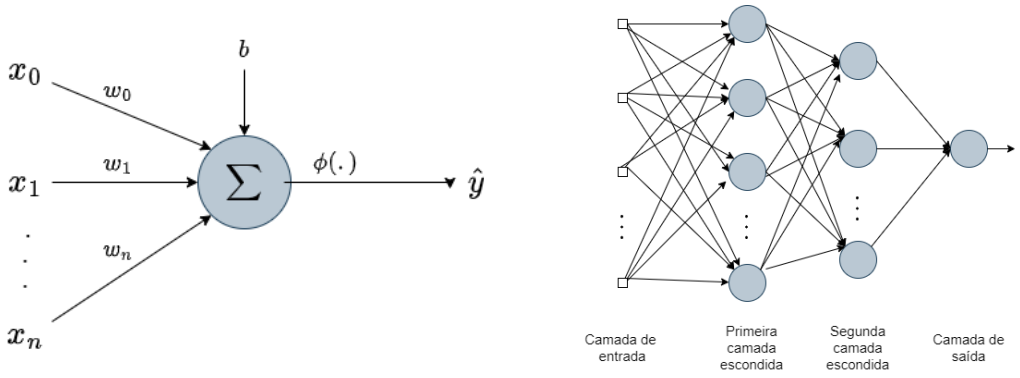
4.5.2. Redes Neurais Artificiais

Redes neurais artificiais (ANN, do inglês: *Artificial Neural Networks*) representam um conjunto de modelos que buscam representar computacionalmente o funcionamento biológico do cérebro humano. Uma ANN é um modelo computacional paralelo e distribuído criado a partir de unidades de computação simples com capacidade de armazenar conhecimento e utilizar o conhecimento adquirido (HAYKIN, 2009). A junção de unidades de computação simples, também conhecidas como neurônios artificiais, formam camadas que são responsáveis por processar e propagar as informações para outras camadas do modelo.

4.5.2.1. Redes Neurais do tipo *Feedforward*

Como apresentado anteriormente, redes neurais artificiais são modelos compostos por camadas de neurônios interconectados entre si. Quando a informação é propagada somente em um sentido do modelo, ou seja, somente para camadas subsequentes, temos uma Rede Neural do tipo *Feedforward* (FNN, do inglês: *Feedforward Neural Network*). O objetivo de uma FNN é aproximar alguma função $f(\mathbf{x}) = y$, de modo que a função possa mapear a entrada $\mathbf{x}$ em uma saída y (GOODFELLOW; BENGIO; COURVILLE, 2016). A unidade de computação simples de uma FNN é o *perceptron*.

Segundo Haykin (2009), o *perceptron* é um modelo computacional formado por uma função afim seguida por um operador não linear. Na Figura 4.5a, podemos observar uma representação gráfica de um *perceptron*. Matematicamente, o *perceptron* é definido conforme a Equação 5, a seguir:



(a) Perceptron

(b) Perceptron de Múltiplas Camadas

Figura 4.5: Representações de um perceptron, unidade básica de processamento de uma FNN (a) e uma MLP, composta de uma camada de entrada, duas camadas escondidas e a camada de saída com um perceptron (b). Adaptado de Haykin (2009).

$$f(\mathbf{x}) = \phi(\sum_{i=0}^{n} w_i x_i + b), \tag{5}$$

onde **x** representa os dados de entrada, **w** é o vetor de pesos, ou conhecimento, associados aos dados de entrada, b representa o viés e ϕ é a função de ativação do *perceptron*. A função de ativação é um operador diferenciável responsável por aplicar, na maioria dos casos, não-linearidade ao modelo (ZHANG et al., 2021). Algumas funções de ativação comumente utilizadas são listadas a seguir:

- Unidade Linear Retificada - ReLU

$$ReLU(x) = max(x, 0); \tag{6}$$

- Sigmóide

$$sigmoide(x) = \frac{1}{1+e^{-x}}; \tag{7}$$

- Tangente Hiperbólica

$$tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}. \tag{8}$$

Apesar de ser descrita como uma unidade de computação simples, ao combinarmos células de perceptrons, podemos gerar estruturas complexas, conhecidas como Perceptron de Múltiplas Camadas (MLP, do inglês: *Multilayer Perceptron*). MLPs são redes neurais do tipo *feedforward* compostas por, no mínimo, três camadas. As camadas de entrada e saída e pelo menos uma camada oculta. Cada camada de uma MLP, com exceção da de entrada, é definida por um conjunto de *perceptrons*. Devido à natureza de uma FNN, cada célula de *perceptron* conecta-se com todas as células de perceptron da camada subsequente, realizando assim a propagação de informação. Na Figura 4.5b é ilustrado a arquitetura básica de uma MLP.

4.5.2.2. Treinamento de Redes Neurais

Como estabelecido anteriormente, algoritmos de aprendizado de máquina adquirem seu conhecimento através de um processo de treinamento. O processo baseia-se na aplicação de métodos de otimização a fim de encontrar os melhores parâmetros para um modelo. Em uma MLP, por exemplo, deseja-se obter o melhor conjunto de pesos $\mathbf{w}$ associados a cada *perceptron* presente nas camadas do modelo. Tal processo é realizado de forma iterativa, ao minimizar ou maximizar uma função objetivo $f(x)$. No contexto de redes neurais, a função objetivo é comumente chamada de função de perda.

Função de Perda: A função de perda tem por objetivo estimar o erro entre a saída $\hat{y}$ de uma rede neural com o valor y real associado aos dados de entrada da rede. Por exemplo, considere um problema de aprendizado supervisionado com objetivo de realizar uma tarefa de predição. Dado o conjunto de amostras $X = \{\mathbf{x}_0, \cdots, \mathbf{x}_N\}$, seus respectivos valores de saída $Y = \{y_0, \cdots, y_N\}$ e os valores de saída estimados pelo modelo $\hat{Y} = \{\hat{y_0}, \cdots, \hat{y_n}\}$, podemos calcular o erro L obtido através de uma função de perda como demonstrado na Equação 9.

$$L(Y, \hat{Y}) = \frac{1}{N} \sum_{i=0}^{N} (y_i - \hat{y}_i)^2 \tag{9}$$

A função de perda demonstrada na Equação 9 é conhecida como Erro Médio Quadrático (MSE, do inglês: *Mean Squared Error*). Outras funções de perda utilizadas em treinamento de redes neurais são:

- Raiz Quadrada do Erro Médio (RMSE)

$$L(Y, \hat{Y}) = \sqrt{\frac{1}{N} \sum_{i=0}^{N} (y_i - \hat{y}_i)^2}; \tag{10}$$

- Erro médio absoluto (MAE)

$$L(Y,\hat{Y}) = \frac{1}{N}\sum_{i=0}^{N}|y_i - \hat{y}_i|; \tag{11}$$

- Entropia Cruzada

$$L(Y,\hat{Y}) = -\sum_{i=0}^{N}(y_i \log \hat{y}_i). \tag{12}$$

A escolha da função de perda está intrinsecamente relacionada ao problema a ser resolvido. Funções como RMSE e MAE são utilizadas para problemas problemas de regressão, quando a saída do problema é um valor numérico. Outras funções de perda, como entropia cruzada, são aplicadas a problemas de classificação.

Otimização por Descida de Gradiente: O objetivo do treinamento de uma rede neural é de minimizar uma função de perda $y = f(x)$ (caso o objetivo seja de maximizar a função, podemos atingir o mesmo objetivo ao minimizar a função $y = -f(x)$). Segundo Goodfellow, Bengio e Courville (2016), ao derivarmos a função $f(x)$ temos uma pequena descida do valor de $f(x)$ no ponto x. Em outras palavras, podemos definir uma variação ε que aplicada a x irá resultar em uma variação correspondente em $f(x)$.

No contexto de treinamento de redes neurais, o objetivo é minimizar a função de perda ao otimizar os valores dos pesos do modelo. Seja $\nabla f(\mathbf{w})$ o gradiente de f. Cada iteração do algoritmo de descida de gradiente irá gerar um novo conjunto de pesos $w^{'}$, definido como:

$$w' = w - \varepsilon \nabla f(\mathbf{w}), \tag{13}$$

onde ε é definido como taxa de aprendizado. Desse modo, a otimização por descida de gradiente consiste em realizar pequenas mudanças do valor de $\mathbf{w}$ em direção oposta ao gradiente, de modo a obter pequenas melhoras nos resultados de saída do modelo.

Backpropagation: Uma MLP pode possuir várias camadas conectadas, ou seja, ela é composta por uma cadeia de funções. Nesse cenário, realizar o cálculo de forma analítica do gradiente pelo algoritmo de descida de gradiente torna-se inviável, uma vez que a operação é computacionalmente custosa e crescente com base no tamanho da rede. Para lidar com esse problema, Rumelhart, Hinton e Williams (1985) propuseram o algoritmo de retro-propagação para o cálculo de gradientes em uma ANN. Inicialmente, o algoritmo realiza os cálculos dos gradientes da rede a partir do erro, obtido através da função de perda, ao avaliar a saída real e a saída predita pela rede. Após essa etapa, é feita a atualização dos pesos da rede. Esse processo é realizado iterativamente até que a rede convirja para uma solução (RUMELHART; HINTON; WILLIAMS, 1985).

4.5.2.3. Redes Neurais Recorrentes

Apesar de FNNs apresentarem bons resultados ao processar dados tabulares, ela perde expressividade ao lidar com dados sequenciais, ou seja, que possuem algum tipo de dependência posicional e/ou temporal. Além disso, dados dessa natureza podem apresentar outras características as quais uma FNN não possui maneiras de processar, como dados de entrada de diferentes tamanhos. Imagine, por exemplo, um problema de processamento de linguagem natural. Sentenças de entrada, assim como frases ditas por pessoas e seu dia a dia, podem ser de tamanhos diferentes. Para lidar com esse tipo de problema, Rumelhart, Hinton e Williams (1986) propuseram uma arquitetura de ANNs, conhecida por Redes Neurais Recorrentes (RNN, do inglês: *Recurrent Neural Networks*).

RNNs são modelos de aprendizado profundo que buscam lidar com características sequenciais existentes em diversos problemas através de conexões recorrentes (ZHANG et al., 2021). Esse processo é realizado através de ciclos dentro da célula de computação simples de uma RNN, que permite o compartilhamento do mesmo conjunto de parâmetros através de uma sequência temporal. Na Figura 4.6, podemos observar a ideia por trás da célula de computação de uma RNN, onde $\mathbf{x}_t$ e h_t representam, respectivamente, a entrada e saída do modelo no tempo t.

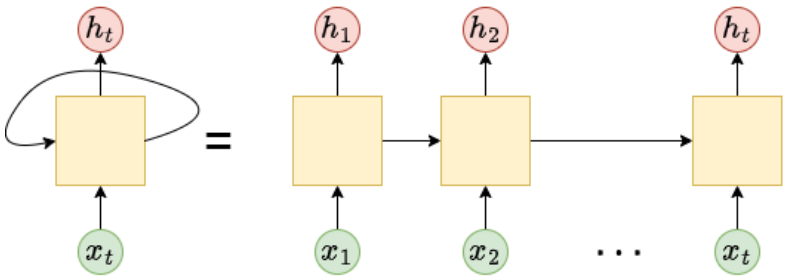


Figura 4.6: Ilustração da célula de computação de uma RNN. Através do ciclo interno presente na célula, é possível manter informações temporais dos dados. Adaptada de Zhang et al. (2021)

Dados utilizados em RNNs são sequências que possuem vetores $\mathbf{x}^{(t)}$ com um intervalo de tempo t variando de 1 a τ. Por exemplo, considere uma série temporal de temperaturas de uma cidade quaisquer aferidas diariamente durante 60 dias. Nesse caso, cada intervalo de tempo t é representado por um dia, ou seja, $\mathbf{x} = \{x^{(1)}, \cdots, x^{(60)}\}$. Agora imagine que queremos prever a temperatura do próximo intervalo de tempo, $x^{(61)}$. É natural supor que a temperatura dos dias anteriores possa influenciar o valor a ser predito. Dessa forma, uma RNN irá utilizar informações do passado, disponível pela natureza de sua arquitetura recorrente, para prever a temperatura do próximo dia. O principal

objetivo de uma RNN é poder utilizar o contexto passado para prever comportamentos futuros.

Apesar de apresentar bons resultados para exemplos em que a dependência temporal dos dados é curta, alguns modelos de RNN não conseguem obter bons resultados quando a dependência temporal dos dados é grande (GOODFELLOW; BENGIO; COURVILLE, 2016). Esse problema é referido como dependência de longo prazo, causado pelo desaparecimento ou explosão do gradiente durante o processo de treinamento de uma RNN (BENGIO; SIMARD; FRASCONI, 1994) e (HOCHREITER et al., 2001). Com o intuito de contornar o problema de dependência de longo prazo de outros modelos de RNN, Hochreiter e Schmidhuber (1997) propuseram um novo modelo conhecido como LSTM (do inglês: *Long Short-Term Memory*). A principal diferença entre a célula LSTM para a célula de uma RNN padrão é a adição de um mecanismo de memória com capacidade de manter informações de estados passados, evitando os problemas de desaparecimento e explosão do gradiente.

Devido aos resultados superiores obtidos por RNNs construídas com LSTM em detrimento de outros modelos, muitos problemas com dados de natureza sequencial passaram a utilizar soluções com aplicação de LSTM. Alguns exemplos de tais aplicações são análises de séries temporais (KARIM et al., 2019) e processamento de linguagem natural (GHOSH et al., 2016). Informações detalhadas da arquitetura de uma LSTM podem ser encotontradas em Hochreiter e Schmidhuber (1997).

4.6. Introdução às Redes Generativas Adversárias

As Redes Generativas Adversárias (GANs) são um *framework* utilizado para a otimização de modelos generativos baseados em aprendizado profundo. A ideia geral do funcionamento de uma GAN pode ser vista na Figura 4.7, que mostra um esquema de falsificação de pinturas. No esquema, o objetivo é treinar um pintor (G) para produzir imagens realistas de um pintor famoso, por exemplo, Pablo Picasso. Para que G tenha conhecimento sobre quão realista estão suas pinturas, o esquema possui também um especialista (D) sobre o estilo de pintura de Picasso. Durante um determinado número de vezes, G produz lotes de pinturas que serão avaliadas por D que, por sua vez, retorna uma nota média que represente o nível de fidelidade das pinturas falsas. Para cada lote, G utiliza um conjunto diferente de cores, garantindo que mesmo que as pinturas sejam idênticas às reais, ainda serão falsas. Assim, considerando que esse processo se repita por um número infinito de vezes, chegará o momento em que D não saberá diferenciar as pinturas falsas das reais, e objetivo do esquema será atingido.

De um ponto de vista técnico, considerando a arquitetura original, as GANs são definidas por um "jogo" de min-max (GOODFELLOW et al., 2014), onde G e D são duas redes neurais treinadas simultaneamente, como pode ser visto na Figura 4.8. A rede G, conhecida como um Gerador ($G(z;\theta_g)$), é um perceptron multicamadas que produz

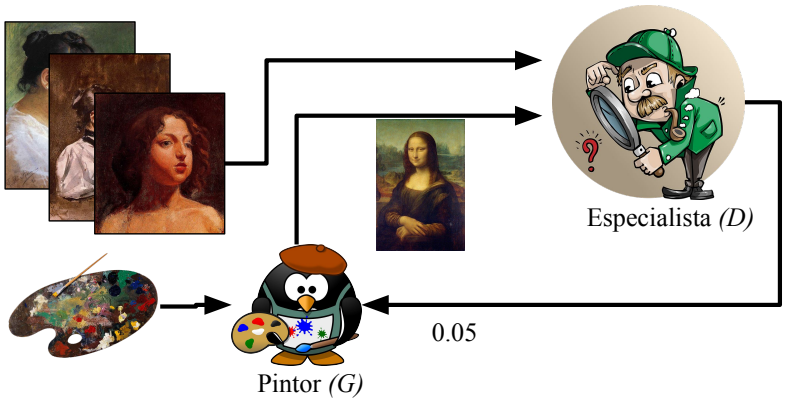


Figura 4.7: Visão geral do funcionamento de uma GAN.

dados falsos com base em entradas aleatórias $p_z(z)$ (as cores do exemplo anterior). D, por sua vez, é o Discriminador ($D(x;\theta_d)$), outro perceptron multicamadas que classifica a qualidade dos dados gerados, considerando uma base de dados reais x. Nas redes, z, θ_g e θ_d significam, respectivamente, o espaço latente dos dados (entradas aleatórias, como distribuições normais), os parâmetros para o perceptron que define G e os parâmetros para o perceptron que define D.

Assim como no exemplo anterior, o objetivo de G é conseguir enganar D. Nesse caso, gerando amostras de dados falsas cuja distribuição $p_g(x)$ se aproxime tanto da distribuição real $p_{data}(x)$, que D não consiga fazer distinção entre elas. A principal diferença entre o exemplo anterior, nesse sentido, é que no início do treinamento, D não pode ser um classificador com uma porcentagem de acurácia muito alta, pois caso isso ocorra, G nunca irá conseguir melhorar a qualidade dos dados gerados. Assim, ao mesmo tempo que D é treinado para distinguir entre os dados oriundos de G e os reais, G terá seus pesos atualizados considerando o *feedback* fornecido por D.

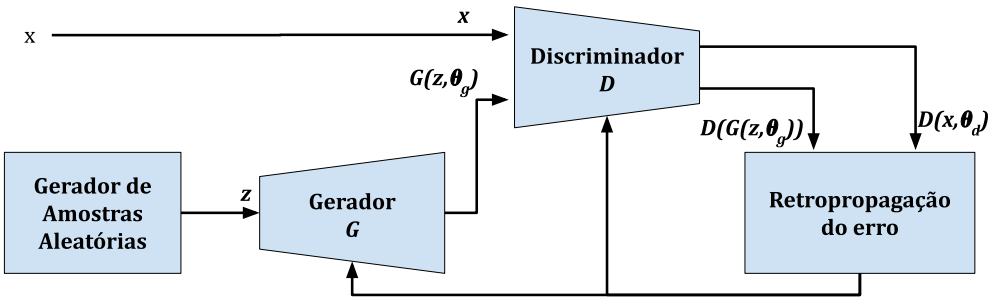


Figura 4.8: Funcionamento de uma GAN. Adaptado de Goodfellow et al. (2014).

Em teoria, como descrito em Goodfellow et al. (2014), ao utilizar a técnica

de min-max e com tempo e recursos computacionais suficientes, espera-se que D e G entrem em equilíbrio. Entretanto, as técnicas usadas para o treinamento de GANs normalmente se baseiam em gradientes descendentes que não são apropriadas para encontrar o equilíbrio do jogo min-max da GAN. Dessa forma, as primeiras GANs poderiam apresentar algumas instabilidades, por exemplo, dada a complexidade dos dados, D e G poderiam nunca chegar a um ponto de convergência. Nesse sentido, o trabalho de Salimans et al. (2016) apresentam uma série de técnicas que podem ser aplicadas para a melhorar o treinamento de GANs.

Figura 4.9: Rostos gerados por uma GAN treinada com a base de dados TFD (SUSSKIND; ANDERSON; HINTON, 2010). Figura de Goodfellow et al. (2014)

Um dos primeiros resultados da aplicação das GANs pode ser visto na Figura 4.9. Os rostos sintéticos foram gerados a partir da base original TFD (SUSSKIND; ANDERSON; HINTON, 2010), uma base de dados com faces de pessoas em baixa resolução e preto e branco. Trabalhos posteriores ao de (GOODFELLOW et al., 2014), utilizando algumas das técnicas propostas por Salimans et al. (2016) em conjunto com novas estruturas de GANs, mostraram a capacidade das GANs de gerarem imagens extremamente realistas, por exemplo, imagens sintéticas com base em diferentes categorias de imagens reais (Figura 4.10) (BROCK; DONAHUE; SIMONYAN, 2018) ou com base em imagens de celebridades (Figura 4.11).

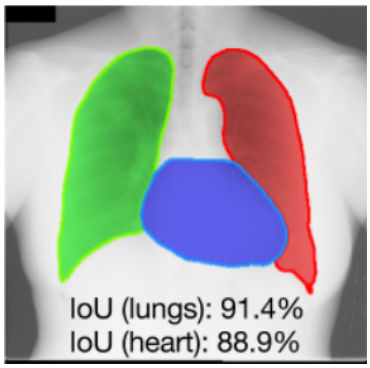
O bom desempenho no campo de visão computacional, principalmente em imagens, chamou a atenção de pesquisadores de diversas áreas. Em medicina, por exemplo, GANs foram utilizadas para na segmentação de órgãos com base em imagens de raios-X (DAI et al., 2018), geração de imagens de lesões em pele (YI; WALIA; BABYN, 2018) e obtenção do fundo de olho a partir de imagens de vasos sanguíneos (COSTA et al., 2017). Esses tipos de dados são fundamentais para o treinamento de modelos classificadores (por exemplo, para identificação de câncer de pele), e a utilização de GANs permite que, a partir de um conjunto pequeno de dados, tenha-se um conjunto maior de dados disponíveis. A Figura 4.12 mostra imagens dos exemplos mencionados. Ressalta-se que

Figura 4.10: Diferentes categorias de imagens geradas pela BigGAN (BROCK; DONAHUE; SIMONYAN, 2018).

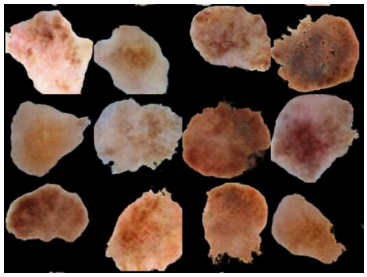
esses são apenas alguns exemplos de aplicações de GANs considerando especificamente o campo de visão computacional. Mais estudos sobre a aplicação em medicina podem ser encontrados nos *survey* de Singh e Raza (2021). Outros estudos sobre GANs em visão computacional podem ser encontrados em Aggarwal, Mittal e Battineni (2021).

Figura 4.11: Rostos gerados com base em imagens de celebridades (KARRAS et al., 2017).

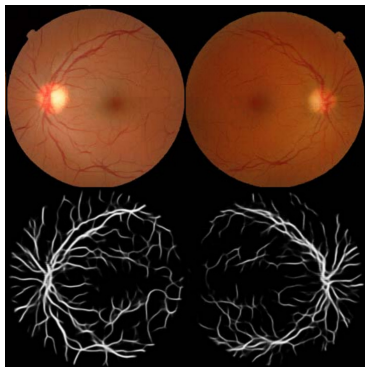
Apesar do seu bom desempenho, a utilização de GANs é desafiadora devido ao seu treinamento "adversário". A questão principal é que, geralmente, não é possível inferir a qualidade dos dados gerados da forma como normalmente é feito para avaliação de modelos de redes neurais, por exemplo, analisando a acurácia ou variação dos valores da função de perda. Do ponto de vista da arquitetura das GANs, um gerador G pode produzir dados que são, estatisticamente, indistinguíveis dos reais para um discriminador D, mas que não fazem sentido para o problema em questão. Isso torna necessário uma avaliação periódica da qualidade dos dados gerados pelos modelos treinados, que pode variar em função do domínio do problema. No caso de dados imagens de rostos, por exemplo, um conjunto de imagens pode ser gerado a cada x iterações a fim de se avaliar



(a) Segmentação de órgãos



(b) Lesões em pele



(c) De vasos para fundo de olho

Figura 4.12: Aplicações de GANs em diferentes áreas da medicina. As figuras foram retiradas dos seus respectivos artigos. (a) Exemplo de segmentação de órgãos (coração e pulmões) à partir de Raios-x (DAI et al., 2018); (b) Diferentes tipo de lesões em pele sinteticamente geradas (YI; WALIA; BABYN, 2018); (c) Obtenção do fundo de olho a partir dos vasos sanguíneos (COSTA et al., 2017).

se os rostos gerados são, de fato, humanos.

Outro desafio diz respeito à construção de um modelo que siga a arquitetura conceitual das GANs. De modo geral, D e G precisam ter capacidades similares, para que nenhuma delas se sobressaia durante o treinamento. Como mencionado anteriormente, D pode ser muito mais poderosa que G e G não consiga evoluir. Da mesma forma, G pode se tornar especialista em gerar um certo tipo de conjunto de dados, pois, pelo *feedback* recebido, sabe que D não consegue diferenciá-lo do real. Ou seja, a distribuição $p_g(x)$ que é aprendida por G foca em poucas variações da distribuição $p_{data}(x)$. Nos exemplos de imagens, ao invés de produzir imagens significativamente diferentes, o modelo acaba gerando imagens com as mesmas características. Esse problema é conhecido como *mode collapse* (YI; WALIA; BABYN, 2019).

Os bons resultados obtidos pelas GANs motivaram pesquisadores de outras áreas a investigar a aplicação das GANs em cenários diferentes do que foi proposto no trabalho original. Nesse sentido, Yu et al. (2017) mostram que a arquitetura original da GAN, com poucas alterações, consegue tratar de dados sequenciais, ainda que com suas limitações. Evidenciando o potencial das GANs para tratar de séries temporais, por exemplo. Trabalhos posteriores evidenciaram que a metodologia proposta nas GANs conseguem ser utilizadas com eficiência para a geração de séries temporais (YOON; JARRETT; SCHAAR, 2019). Contudo, ainda há diversos desafios inerentes à arquitetura das GANs que podem dificultar sua utilização em cenários de séries temporais ainda não

estudados. Alguns desses desafios, bem como oportunidades de pesquisa, serão descritas na próxima seção.

4.7. Desafios e Oportunidades de Pesquisa

Nessa seção, discutiremos alguns problemas e desafios de pesquisa acerca da utilização de GANs para geração de séries temporais. Leitores mais interessados podem se aprofundar nos detalhes em (LIN et al., 2020) e (ZHANG; MA; XIA, 2022).

4.7.1. Implementação, Treinamento e Avaliação

Um dos principais desafios da utilização de GANs, de uma maneira geral, é encontrado já durante sua implementação, principalmente em relação às capacidades dos modelos e função de perda utilizada. No primeiro caso, considerando a arquitetura original, o gerador e o discriminador precisam ter capacidades equivalentes. Caso um dos componentes seja muito mais poderoso que o outro, pode ocorrer o problema de *model collapse*, citado anteriormente. A função de perda é outro fator importante, pois é ela que dá o *feedback* que faz com que o gerador melhore a qualidade dos dados gerados.

A avaliação dos modelos é outro fator importante durante a utilização de GANs. Um dos problemas recorrentes na literatura são trabalhos que propõe modelos específicos para um campo de estudo, mas testados em apenas uma base de dados, como modelos para geração de música (ENGEL et al., 2019) e sinais de eletroencefalografia (HARTMANN KAY GREGOR E SCHIRRMEISTER; BALL, 2018). Tais modelos, apesar de eficientes, precisam ser avaliados em outras bases de dados. Além disso, em muitas situações a adequação dos dados sintéticos aos problemas reais necessitam da avaliação "manual" de um humano, o que pode tornar o processo de treinamento ainda mais trabalhoso. Modelos de GANs para geração de músicas (ENGEL et al., 2019) e dados de mobilidade (RIBEIRO et al., 2021b) são exemplos onde a avaliação manual é necessária para a definição de um bom modelo. Um grande desafio, nesse sentido, é encontrar formas automatizadas de se realizar esse tipo de avaliação.

Por fim, é importante ressaltar os problemas que envolvem modelos de aprendizado profundo, os componentes principais das GANs. No geral, o treinamento desses modelos envolve um processo que demanda tempo e complexo, em um longo exercício de tentativa e erro, já que diversos aspectos devem ser considerados, como hiperparâmetros, algoritmos de otimização (e seus hiperparâmetros), entre outros (RAMOS et al., 2021). Assim, a busca por um bom modelo de GAN também pode encontrar esses mesmos problemas e técnicas para buscas automáticas de modelos se fazem necessárias. Nesse sentido, já é possível encontrar essas técnicas considerando aplicações de visão computacional (GAO et al., 2020), mas a área de séries temporais ainda carece de estudos nessa direção.

4.7.2. Generalização

Uma das grandes características das GANs para visão computacional é que um mesmo modelo pode ser utilizado, sem exigir grandes modificações, para problemas similares. Por exemplo, a GAN original, o modelo mais simples já definido, obteve resultados satisfatórios em duas bases de dados semanticamente diferentes (rostos de pessoas e dígitos) mas conceitualmente as mesmas, pois se tratavam de imagens. Assim, como sendo o foco das GANs desde o início, nota-se que há arquiteturas e configurações específicas para problemas de visão computacional.

Para séries temporais, contudo, a obtenção de um modelo generalista encontra desafios adicionais. O principal fator é que uma série temporal não se limita a conter dados de um único tipo. Por exemplo, dados de mobilidade urbana podem ser representados por uma série temporal e conter dados numéricos (latitude, longitude, altitude, velocidade) e categóricos (tipo de local visitado, tipo de trajetórias) (RAO et al., 2020). Outro exemplo são dados de pacientes de um hospital, que podem conter informações numéricas (leituras feitas por aparelhos eletrônicos) ou categóricas (rótulos associados aos pacientes) (ESTEBAN; HYLAND; RÄTSCH, 2017). Nesse contexto, em situações práticas pode ser necessário que se encontre formas eficientes de se tratar esses dados simultaneamente, o que pode adicionar ainda mais complexidade à tarefa.

Além disso, na literatura é comum encontrar problemas similares sendo abordados por metodologias diferentes, onde cada metodologia possui suas vantagens e desvantagens que devem ser considerados durante sua aplicação. Por exemplo, em relação à geração de músicas, (MOGREN, 2016) propôs uma arquitetura similar à GAN clássica, onde o gerador e discriminador são RNNs. Por outro lado, (YU et al., 2017) propõe uma arquitetura baseada em aprendizagem por reforço, a qual é uma metodologia diferente do que é geralmente utilizada em GANs. Neste exemplo, caso deseje-se comparar os dois modelos, será necessário compreender bem as duas formas de aprendizado de máquina que podem ser bem distintas. Essas diferenças tornam-se maiores quando se trata de áreas diferentes, por exemplo, considerando a área de mobilidade urbana. Alguns modelos para geração de mobilidade combinam RNNs e Redes Convolucionais (JAUHRI et al., 2020).

4.7.3. Privacidade

Por definição, as GANs já fornecem privacidade aos dados sintéticos gerados, pois cada geração depende de entradas aleatórias. Entretanto, dado que as GANs conseguem capturar as características dos dados reais, diferentes questões de privacidade surgem a partir da aplicação de GANs em diferentes áreas. Em (LIN et al., 2020) os autores apresentam diversos aspectos sobre privacidade e GANs:

4.7.3.1. Proteção de segredos empresariais

Uma das grandes vantagens das GANs é permitir o compartilhamento de dados sintéticos ou modelos que gerem dados sintéticos. Segundo (LIN et al., 2020), uma das preocupações de quem detém os dados é o vazamento de informações sigilosas como recursos disponíveis e em uso na empresa, que estão geralmente embutidas em metadados. Contudo, para a realização de estudos, por exemplo, simulação de dados de uma aplicação, pode ser interessante que as correlações entre medições reais da aplicação e esses metadados sejam mantidas.

As alternativas para minimizar esse problema são mudar ou ofuscar a distribuição dos metadados. Nesse caso, (LIN et al., 2020) propõe uma arquitetura para realizar esse tipo de procedimento, em que medições reais e metadados são gerados por modelos independentes. Assim, após o treino nos dados reais, retreina-se o modelo dos metadados para geração de distribuições desejadas. A questão principal e maior interesse de pesquisa é como otimizar a distribuição dos atributos, já que o treinamento, nestes casos, funciona a partir de meios condicionais, isto é, a geração dos metadados é condicionada, probabilisticamente, por outras variáveis de entrada do modelo.

4.7.3.2. Proteção da privacidade de usuários

Com a criação da LGPD (BRASIL, 2018), a preocupação com a proteção e manutenção da privacidade dos dados de usuários intensificou-se em diversas áreas que envolvem aplicações que utilizam ou compartilham dados de usuários. Como que o objetivo principal de se utilizar GANs é reproduzir as características dos dados reais, tem-se um desafio enorme em equilibrar a geração de dados sintéticos e manutenção da privacidade. Um exemplo de como a privacidade de um usuário pode ser comprometida ocorre quando um modelo generativo memoriza as características de um indivíduo específico e vaza essas informações durante a geração dos dados (CARLINI et al., 2019).

Existem diversas formas de tratar destas questões e em muitas delas o principal desafio é como manter o equilíbrio entre utilidade e privacidade dos dados, isto é, quanto mais privacidade fornecida aos dados, menos úteis eles tendem a ser. Nesse contexto, uma das métricas mais utilizadas para avaliar a privacidade de usuários é a privacidade diferencial (DWORK, 2008). Essa métrica define que um modelo não deveria depender dos dados de um usuário específico de forma que, no contexto de modelos generativos, os dados sintéticos representem as características gerais dos dados. Na literatura, diversos estudos propuseram modelos de GANs para séries temporais considerando a privacidade diferencial (FRIGERIO et al., 2019), por adição de ruídos no processo de treinamento (principalmente durante a aplicação da descida do gradiente). Mais recentemente (QU et al., 2020) utilizaram esta técnica em dados de mobilidade, e

mostraram o que mencionamos anteriormente: quanto mais privacidade era fornecida aos dados, menos características dos dados reais era aprendida. Além disso, segundo (LIN et al., 2020), a utilização da privacidade diferencial ainda carece de estudos que avaliem os dados, principalmente em relação à qualidade dos dados.

Outra técnica utilizada para quantificar a privacidade é através dos ataques de inferência de membro (*membership inference attack*) (CHEN et al., 2020). O objetivo deste ataque é, considerando um conjunto de amostra de dados e um modelo de aprendizado de máquina, inferir se tal conjunto faz parte dos dados de treino. Nesse sentido, um modelo em que a privacidade diferencial tenha sido corretamente aplicado reduziria as chances de sucesso desse tipo de ataque. Além disso, em seu estudo utilizando a base de dados *Wikipedia Web Traffic* (GOOGLE, 2018), (LIN et al., 2020) verificaram que as chances de sucesso do ataque podem ser reduzidas quando se utiliza uma base de dados maior durante o treino.

4.8. Estudo de Caso - Hands-on

Para pôr em prática os conceitos aprendidos durante o capítulo, nesta seção apresentamos uma aplicação de geração de séries temporais utilizando GANs. A base de dados utilizadas representa o número de bicicletas alugadas em um serviço de locação em 7 cidades dos Estados Unidos e estão disponíveis um repositório no Github[1]. A Figura 4.13 apresenta a metodologia que será seguida nas próximas subseções. Primeiramente, faremos a análise e descrição dos dados (Subseção 4.8.1), seguido das modelagens estatísticas (Subseção 4.8.2) e com GANs (Subseção 4.8.3). Por fim, apresentamos como os modelos podem ser avaliados (Subseção 4.8.4).

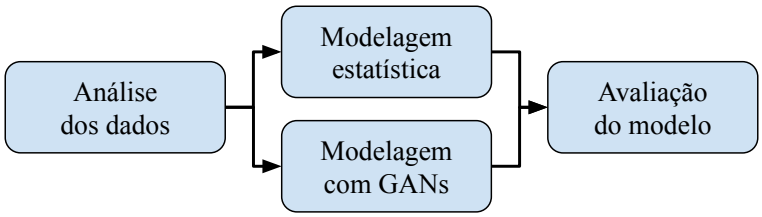


Figura 4.13: Metodologia para geração de séries temporais

4.8.1. Análise e Descrição dos dados

A base de dados possui 78888 registros de bicicletas alugadas, entre janeiro de 2011 e dezembro de 2019. Cada valor representa o número de bicicletas alugadas simultaneamente em intervalos de 1 hora durante cada dia entre 2011 e 2019. Como um dia possui 24 horas, a base de dados pode ser representada com um vetor com 3287 registros, onde

[1]https://github.com/ifribeiro/minicurso_webmedia22

cada registro possui 24 linhas (cada linha representa uma hora) e cada linha armazena uma variável (o número de bicicletas daquela hora). A Tabela 4.1 mostra as 5 primeiras linhas das três colunas da base de dados. Nosso interesse aqui é apenas na coluna *cnt*, mas as colunas *date* e *hour* auxiliam na visualização e entendimento dos dados.

Tabela 4.1: Primeiras 5 linhas da base de dados *Bikesharing*

date	**hour**	**cnt**
2011-01-01	0	16
2011-01-01	1	40
2011-01-01	2	32
2011-01-01	3	13
2011-01-01	4	1

Ao longo do minicurso, utilizaremos uma série de bibliotecas que implementam boa parte das funcionalidades de que necessitamos, principalmente as bibliotecas dos modelos ARIMA e do modelo de GAN utilizado e bibliotecas de visualizações de dados. Para facilitar a execução, algumas funções auxiliares de visualização foram implementadas e estão disponíveis no repositório. As versões das bibliotecas utilizadas são destacadas no git.

Assim, inicialmente carregamos a base de dados utilizando a biblioteca pandas e visualizamos todos os dados com o código a seguir:

```
bikes = pd.read_csv('datasets/bike_sharing_2011to2019.csv')
lista_datas = p.get_list_dates(data_size=bikes.shape[0],
year=2011, month=1, day=1)
fig, ax = plt.subplots(figsize=(10,5))
ax.plot(lista_datas, bikes['cnt'])
ax.set_xlabel("Data")
ax.set_ylabel("Numero de bicicletas")
```

A Figura 4.14 mostra a série temporal representando o número de bicicletas alugadas nas 7 cidades de onde os dados foram coletados entre janeiro de 2011 e dezembro de 2019. Nota-se que há uma tendência de crescimento no número de bicicletas alugadas entre 2011 e início de 2019. Além disso, em cada ano os dados apresentam características similares (sazonais) em que há menos bicicletas alugadas no início do ano, com uma tendência de crescimento até por volta do meio do ano e, por fim, uma tendência de redução até o final do ano.

Apesar de fornecer informações importantes sobre os dados, a Figura 4.14 dificulta a visualização de informações que podem auxiliar no melhor entendimento dos dados, como pode ser visto na Figura 4.15. Na imagem à esquerda, tem-se o primeiro

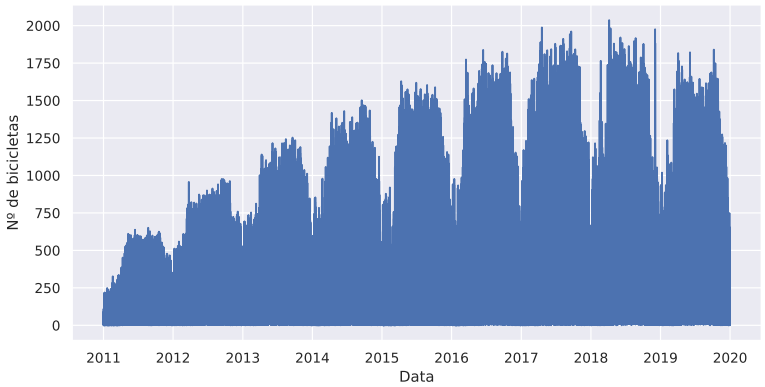


Figura 4.14: Número de bicicletas alugadas entre Janeiro de 2011 e Dezembro de 2019

mês dos dados. Nota-se que existe uma sazonalidade semanal dos dados, com uma sequência de valores altos e baixos. Além disso, podemos olhar especificamente para a primeira semana dos dados (imagem à direita), e notar que cada dia apresenta propriedades específicas. Por exemplo, em 01/01 e 02/01 (dois primeiros dias) os dados apresentam um pico por volta meio-dia. Se olharmos no calendário de 2011, esses dias correspondem ao sábado e domingo, respectivamente. Nos 5 dias subsequentes (dias úteis), como pode ser visto, os dados possuem um pico no início e no fim do dia, que variam pouco de um dia para o outro.

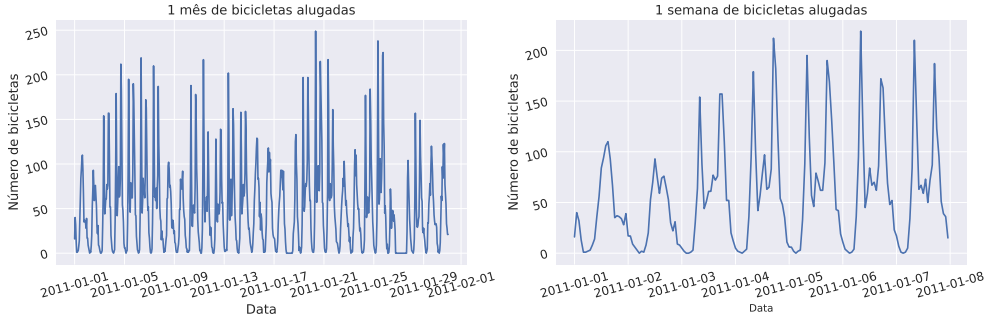


Figura 4.15: Um mês (figura à esquerda) e uma semana (figura à direita) de bicicletas alugadas.

Apesar de já entendermos melhor parte dos dados, apenas olhar para as duas figuras anteriores, não seria o suficiente para entendê-los como um todo. Nesse sentido, a Figura 4.16 apresenta a sumarização dos dados por semana (aqui denominados de soma por intervalo), em que computamos o número de bicicletas alugadas por hora do dia,

considerando os valores acumulados de cada dia da semana. A Figura evidencia o que foi discutido anteriormente, mostrando que há uma diferença clara entre os dias úteis e finais de semana. Além disso, conseguimos identificar que os dois picos dos dias úteis são, respectivamente, 9 da manhã e 6 da tarde.

```
real_data = bikes["cnt"].values.reshape(bikes.shape[0]//24, 24)
df_real = p.get_df(list_dates=lista_datas,
                   data=bikes[["cnt"]].values, timesteps=24)
lista_cnt = [p.get_count(df_real,w,24,
                   column="ts")["cnt_0"] for w in range(7)]
p.plot_sum_real(list_cnt_real=lista_cnt, list_dates=lista_datas)
plt.ticklabel_format(style="sci", axis="y", scilimits=(0,0))
plt.tight_layout()
```

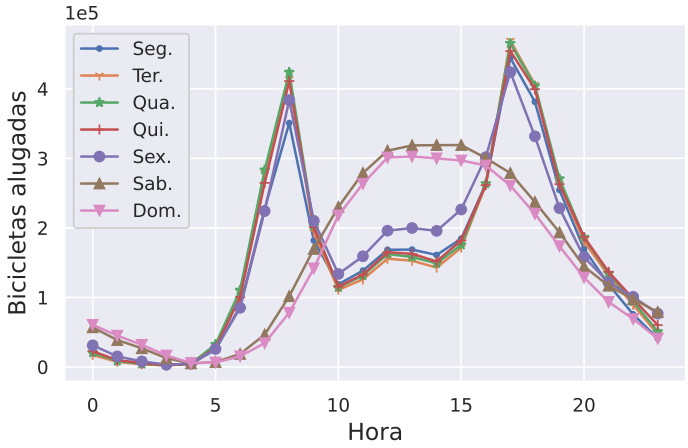


Figura 4.16: Somas por intervalo para a base de dados *Bikesharing*

As análises que fizemos até aqui serão de grande importância para entender como melhor definir os modelos que serão utilizados na geração dos dados sintéticos, bem como as causas de possíveis problemas que podemos encontrar durante as modelagens. Nas próximas subseções, apresentamos as técnicas de modelagem de dados.

4.8.2. Modelagem estatística

Para mostrar a diferença entre as GANs e modelos lineares, apresentamos aqui uma modelagem dos dados utilizando o ARIMA. Esta etapa é importante para verificarmos a necessidade ou não de se utilizar modelos mais complexos. Por exemplo, em outras situações, a utilização de um modelo baseado em aprendizado profundo pode ser desnecessário caso um modelo ARIMA já produza os resultados desejados.

Como mencionamos anteriormente, por ser um modelo baseado em regressão, o ARIMA faz algumas suposições sobre os dados, a principal é que eles sejam estacionários (a média não varia significativamente em função do tempo). Caso a série temporal não seja estacionária, precisamos aplicar diferenciações nos dados, de forma que a série resultante passe a ser estacionária. Há diversas formas de se verificar isso, mas neste minicurso utilizamos uma biblioteca do Python que já possui as ferramentas necessárias.

Devido à grande quantidade de dados, alguns códigos podem demorar alguns minutos para finalizar a execução, dependendo do poder computacional disponível. Assim, neste estudo de caso, utilizaremos apenas o primeiro ano dos dados reais a partir desta subseção. Ressaltamos que isso não altera a forma como os dados são executados, apenas o tempo de execução. Desta forma, primeiramente verificamos se os dados são estacionários com o código a seguir.

```
# realiza teste de estacionariedade
adf_test = pm.arima.stationarity.ADFTest(alpha=0.05)
treino = bikes["cnt"].iloc[:365*24].values
p_value, dif = adf_test.should_diff(treino)
print(p_value, dif)
```

No teste acima se a série temporal deve passar por alguma tipo de diferenciação. Nesse caso, o valor da variável `dif` pode ser *False*, caso não seja necessário a diferenciação e *True*, do contrário. Em nosso estudo de caso, o teste retorna que os dados são estacionários, logo nenhuma diferenciação precisa ser feita. Esse resultado já nos informa que o parâmetro d do ARIMA será igual a 0.

Em seguida, verificaremos quais os valores para os outros dois parâmetros, p e q, do modelo. Anteriormente apresentamos a metodologia de Box & Jenkins, que poderia ser utilizada para estimar tais parâmetros, mas utilizaremos uma biblioteca que automatiza esse processo com uma metodologia similar. Para isso, definiremos quais os valores possíveis de cada parâmetro para que a biblioteca verifique qual combinação deles se ajusta melhor aos dados. Assim, no trecho de código abaixo, definimos que p e q iniciam-se em 0 (`star_p=0`, `start_q=0`) e terminam em 3 (`max_p=3`, `max_q=3`). Além disso, o teste é executado 5 vezes (`maxiter=5`). Nesse caso, valores maiores para o parâmetro `maxiter` garantem maior confiabilidade no resultado obtido.

Desta forma, no código abaixo verificamos quais parâmetros geram um modelo que se ajusta melhor aos dados. É importante ressaltar que os valores (p, d e q) encontrados podem variar um pouco dependendo dos parâmetros passados à biblioteca e devido à natureza probabilística da sua implementação. Neste exemplo, a biblioteca retorna que os melhores parâmetros são: $p = 3$, $d = 0$ e $q = 1$. Com esses valores, ajustamos um modelo ARIMA aos dados, com as duas últimas linhas do código abaixo.

```
# verifica melhores parâmetros para o modelo (d=0, por padrão)
pm_auto = pm.auto_arima(treino, maxiter=10, start_p=0, max_p=3,
                        start_q=0, max_q=3, stationary=True)
# ajusta um modelo usando os parametros identificados
arima = ARIMA(treino, order=pm_auto.order)
res = arima.fit()
```

4.8.3. Definição e treinamento das GANs

Após a definição do modelo ARIMA, podemos definir o modelo GAN. Como mostramos na Subseção 4.8.1, os dados variam entre valores muito pequenos e muito grandes, o que pode prejudicar o processo de treinamento. Assim, antes da definição dos parâmetros é preciso padronizar os dados para que eles estejam em uma escala menor, por exemplo, no intervalo $[0,1]$. Isso pode ser feito com o código a seguir:

```
scaler = MinMaxScaler().fit(treino.reshape(-1,1))
treino_scaled = scaler.transform(treino.reshape(-1,1))
treino_s = treino_scaled.reshape(len(treino)//24, 24, 1)
```

Nesse caso, é importante ressaltar o tipo de entrada esperada pela GAN. Como utilizamos um modelo baseado em séries temporais, a entrada esperada precisa ter 3 dimensões (n,m,v), onde o n é o número de registros, m é o tamanho da sequência em cada registro e v e o número de variáveis. Em nosso caso, a variável `treino_s` do código anterior tem as dimensões $(365,24,1)$.

Nesse sentido, a GAN que utilizamos neste estudo de caso possui diversos parâmetros para serem configurados e alguns deles têm uma influência maior nos resultados e outros no treinamento. Discutiremos isso com mais detalhes na próxima subseção. O importante, nesta etapa, é entender a escolha de alguns parâmetros com base na análise feita na Subseção 4.8.1, especificamente: `seq_len`, `n_seq`, `hidden_dim`, `batch_size` e `learning_rate`. Nesse sentido, os dois primeiros parâmetros são obtidos diretamente das duas últimas dimensões dos dados. Assim, `seq_len=24` e `n_seq=1`.

Definimos também o parâmetro `hidden_dim=24`, referente ao número de unidades em cada camada escondida das redes neurais que compõe a GAN em que conseguimos controlar a capacidade de aprendizado do modelo. Como temos uma quantidade relativamente pequena de dados, esse valor não precisa ser muito grande. Em seguida, definimos `batch_size=28`, que é o tamanho do lote de dados utilizado durante cada iteração do treinamento que, em nosso problema, é a quantidade de dias utilizados para treinar o modelo. Geralmente o `batch_size` é um valor razoavelmente representativo dos dados, que possibilite que o modelo aprenda e garanta que o modelo

irá receber valores diferentes de dados à cada iteração. Como vimos, na etapa de análise, que os dados apresentam um padrão semanal, optamos por definir esse parâmetro com o valor 28, ou seja, o modelo recebe, aproximadamente, um mês de dados a cada etapa do treinamento. Por fim, definimos o parâmetro `learning_rate=5e-4 (0.00005)`, que controla a velocidade com que os modelos aprendem as características dos dados. Após a definição dos parâmetros, chamamos a função `ModelParameters` que salva os parâmetros na forma esperada pela biblioteca utilizada.

```
seq_len=24
n_seq = 1
hidden_dim=24
batch_size = 28
learning_rate = 5e-4
gamma=1
dim = 128
noise_dim = 32
gan_args = ModelParameters(batch_size=batch_size,
                           lr=learning_rate,
                           noise_dim=noise_dim,
                           layers_dim=dim)
```

Em seguida, podemos realizar o treinamento do modelo. No código do exemplo abaixo definimos que o modelo será treinado por 3000 vezes. É importante ressaltar que esse valor pode influenciar bastante o tempo de treinamento, principalmente se a máquina utilizada não possui recursos suficientes. Por exemplo, utilizando a versão grátis do Google Colab[2], o código abaixo demora cerca de 1h para finalizar. Contudo, caso seja de interesse do leitor, um modelo já treinado encontra-se salvo no repositório do minicurso. Após o treinamento, podemos salvar o modelo com o código da linha 3 para uso futuro.

```
df_sample = p.get_df(lista_datas[:365*24], treino.reshape(-1,1 ),
                     timesteps=24)
dict_weeks_real_arima = {wk:p.get_count(df_sample,wk,timestep=24,
                     column="ts")["cnt_0"] for wk in range(7)}

path_lista_fakes = glob.glob("datasets/generated/arima/*.npy")
dict_weeks_fakes = {i:p.get_list_wks(path_lista_fakes,
                     lista_datas[:365*24], 24, wk=i) for i in range(7)}
p.plot_compare_sum(dict_weeks_fakes, dict_weeks_real_arima,
                     bbox=(1.45, -0.1), figtitle="", scaler=None)
```

[2]https://colab.research.google.com/

4.8.4. Avaliação dos modelos

Com os modelos treinados, podemos gerar os dados sintéticos e verificar o desempenho dos modelos em realizar a tarefa desejada. Ressalta-se que um bom modelo terá dados que mantenham as principais características dos dados e em que cada base sintética gerada seja diferente. Assim, para realizar as avaliações, primeiro geramos os dados sintéticos com o modelo ARIMA e com o modelo GAN. Para garantir os resultados obtidos, é interessante que seja gerada uma quantidade razoável de dados sintéticos. No exemplo apresentado aqui, vamos gerar e salvar 10 bases para cada modelo:

```
# geração de 10 bases sintéticas com o modelo arima
for i in range(10):
    synth_arima = res.simulate(nsimulations=len(treino))
    np.save('datasets/generated/arima/arima_{}.npy'.format(i),
            synth_arima)
# Geração de 10 bases sintéticas usando GANs
for i in range(10):
    synth_data = synth.sample(len(treino_s))
    np.save("datasets/generated/timegan/timegan_{}.npy".format(i),
            synth_data)
```

Com as bases geradas, podemos realizar as avaliações. Nesse sentido, realizamos dois tipos de avaliação. Na quantitativa, verificamos os resíduos produzidos pelos modelos, que permite avaliar, ao mesmo tempo, se os dados sintéticos mantém as principais características dos modelos e se possuem variação. Na avaliação qualitativa nosso foco é verificar se os dados gerados são de fato similares aos reais e em quais pontos o modelo pode ser melhorado.

4.8.4.1. Avaliação quantitativa

A avaliação quantitativa é feita usando os erros residuais dos modelos treinados. Os erros residuais e de um modelo consistem da diferença entre o que se espera de saída de um modelo (representado por y) e o que o modelo retorna (representado por $\hat{y}$), ou seja, $e = y - \hat{y}$. A ideia de se utilizar os erros residuais é que, supondo que os dados sintéticos tenham capturado as características dos reais, seria possível utilizar um modelo de previsão para compará-los. Por exemplo, considere que um modelo tenha gerado um conjunto de dados sintéticos para uma série temporal com 200 registros. Se um modelo de previsão tenha sido treinado utilizando os 100 primeiros registros dos dados reais, a previsão resultante para os 100 registros restantes deve possuir características similares aos 100 últimos registros dos dados sintéticos.

Um modelo que tenha capturado com eficiência as propriedades dos dados reais, terá resíduos próximos de 0. Assim, caso o modelo anterior tivesse gerado a saída ($\hat{y}$)

exatamente como esperado (y), tanto a média quanto o desvio padrão seriam 0. Entretanto, o ideal é que os dados sintéticos gerados pelos modelos possuam variabilidade, ou seja, cada base é diferente entre si e, ao mesmo tempo, possuam as propriedades dos dados reais. Em outras palavras, os resíduos devem ser um ruído de média zero e variância constante.

Nesse sentido, a Figura 4.17 apresenta os erros residuais nas situações em que as séries temporais previstas pelo modelo são iguais (os dados sintéticos são iguais aos reais) e quando são levemente diferentes. No primeiro caso, os erros são todos iguais a 0, bem como a média e desvio padrão, contudo, isso significa que não temos dados sintéticos, e sim uma cópia dos dados reais. No segundo caso, ao calcularmos a média e desvio padrão obtemos valores diferentes de 0, mas notamos que os dados são similares.

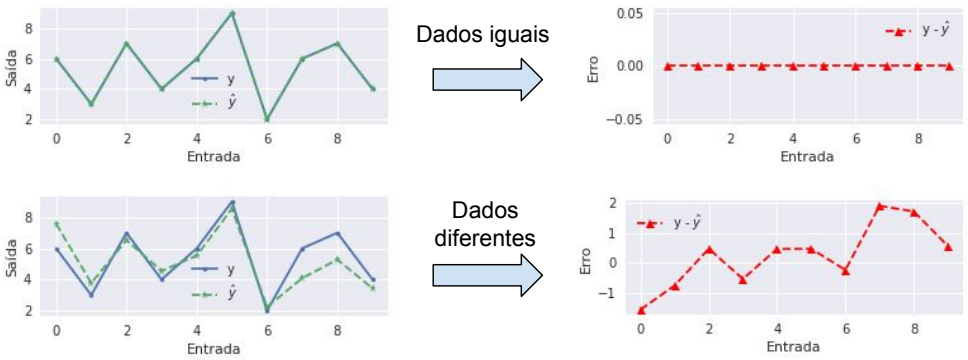


Figura 4.17: Exemplo de erros residuais quando os dados sintéticos e reais são iguais e diferentes

Em nosso exemplo, a obtenção dos resíduos precisa ser feita com os modelos adequados para cada situação. Assim, os erros residuais do ARIMA são obtidos diretamente do modelo ajustado aos dados e os erros residuais das GAN de um modelo de rede neural recorrente. Primeiramente definimos a rede neural recorrente que servirá como modelo de previsão para geração dos resíduos. Para isso, utilizamos a função `make_rnn_model` disponível no código `residuals.py` do repositório. No exemplo de código abaixo, nas linhas de 1 a 3 definimos e compilamos a rede neural. Nas linhas 4 e 5 transformamos os dados de treino no padrão esperado pela RNN. O treinamento, neste caso, consistem em prever, a partir de 24 registros (1 dia), quais os próximos 24 registros. Por fim, na linha 7 treinamos a rede nos dados reais, definindo, com a função `EarlyStopping`, que o treinamento será interrompido caso não haja melhoras significativas no valor retornado pela função de perda.

```
rnn_model = r.make_rnn_model(units=32, n_layers=2, net_type="lstm")
opt = Adam(learning_rate=5e-4)
rnn_model.compile(optimizer=opt, loss="mse")
X, Y = r.split_sequence(treino_s.flatten(),24)
X = X.reshape(X.shape[0],X.shape[1],1)
early_stop = EarlyStopping(monitor="loss")
hist=rnn_model.fit(X,Y, epochs=50, batch_size=28,
            callbacks=[early_stop])
```

Com o modelo treinado, podemos obter os resíduos. Primeiramente, listamos os nomes dos arquivos gerados pela GAN (linha 1). Em seguida, usando o modelo treinado, prevemos qual os dados obteríamos caso a entrada fosse os dados reais (linha 3) e, na linha 4, obtemos os resíduos de cada dado sintético com a função `get_residuals`. Nesse sentido, o mesmo processamento feito nas linhas 4, 5 do código anterior é feito para cada dado sintético. Assim, o que é esperado pelo modelo (`X_pred`) e o que o modelo gera podem ser utilizados para obtenção dos resíduos. A partir dos cálculos dos resíduos podemos calcular e armazenar as média e o desvio padrão, nas linhas de 6 a 9 do código abaixo.

```
synth_m3 = glob.glob("datasets/generated/timegan/3/*.npy")
X_pred = rnn_model.predict(X)
reid_m3 = [r.get_residuals(f, X_pred) for f in synth_m3]
dict_resid = {
    "mu": [ np.mean(resid_arima), np.mean(reid_m3)],
    "std": [np.std(resid_arima), np.std(reid_m3)]}
df_resid = pd.DataFrame(dict_resid, index=["ARIMA", "GAN"])
```

A Tabela 4.2 mostra os resultados dos resíduos de cada modelo. Nota-se que no geral a média de ambos os modelos são muito próximas de 0, entretanto o ARIMA tem um desvio padrão muito maior do que o modelo GAN. Isso é um indicador muito forte de que o ARIMA não conseguiu gerar dados com as características principais dos dados reais. Para confirmar isso, é importante que verifiquemos a análise qualitativa a seguir.

	Média	Desvio padrão
ARIMA	-0.009	64.608
GAN	-0.049	0.276025

Tabela 4.2: Média e desvio padrão residual dos modelos

4.8.4.2. Avaliação qualitativa

Na análise qualitativa, o que fazemos é basicamente gerar um resultado similar ao da Figura 4.16, que é a soma por intervalo, mas considerando as médias e desvios padrão das somas. Assim, primeiramente calculamos as somas, para cada dia da semana, de cada base sintética, e obtemos a média a desvio padrão. Por fim, comparamos esses valores com a soma por intervalo dos dados reais do dia da semana correspondente.

O código abaixo realiza esse cálculo para o modelo ARIMA. Primeiramente, obtemos as somas dos dados reais nas linhas 1 e 2. Em seguida, das linhas 4 a 6, calculamos as somas por intervalo dos dados sintéticos e geramos a Figura 4.18 que compara os resultados. Na Figura, a linha azul corresponde aos dados reais e as linhas de outras cores à média das somas por intervalo dos dados sintéticos. A parte sombreada em cada linha colorida corresponde ao desvio padrão dos dados sintéticos e indica como cada base de dados varia em cada dia da semana.

```
1 df_sample = p.get_df(lista_datas[:365*24], treino.reshape(-1,1),
2                      timesteps=24)
3 dict_weeks_real_arima = {wk:p.get_count(df_sample,wk,timestep=24,
4                  column="ts")["cnt_0"] for wk in range(7)}
5 path_lista_fakes = glob.glob("datasets/generated/arima/*.npy")
6 dict_weeks_fakes = {i:p.get_list_wks(path_lista_fakes,
7                  lista_datas[:365*24], 24, wk=i) for i in range(7)}
8 p.plot_compare_sum(dict_weeks_fakes, dict_weeks_real_arima,
9                  bbox=(1.45, -0.1), figtitle="", scaler=None)
```

A partir da figura anterior, fica evidente que o ARIMA não conseguiu capturar as características dos dados, como foi esperado a partir da análise quantitativa. Assim, nesse caso, o ideal é buscar por outros modelos que apresentem um desempenho melhor.

Para avaliar a GAN, realizamos um processo muito similar ao anterior. A diferença aqui é que utilizamos os dados reais que estão na escala [0, 1] para realização das análises, já que os dados gerados pela GAN também estão neste intervalo. Assim, a Figura 4.19 mostra as somas por intervalos das GANs geradas a partir das 10 bases sintéticas. Fica evidente que, diferente do ARIMA, a GAN conseguiu capturar com eficiência as principais características dos dados reais: os picos de bicicletas alugadas às 8 e às 18 horas, bem como o leve aumento de bicicletas alugadas entre 12 e 13 horas. Além disso, as áreas sombreadas mostram que os dados sintéticos possuem uma boa variabilidade. Nesse sentido, ressaltamos que a Figura está em um escala reduzida. No geral, a área sombreada indica que há uma diferença de cerca de 100 bicicletas entre cada base sintética.

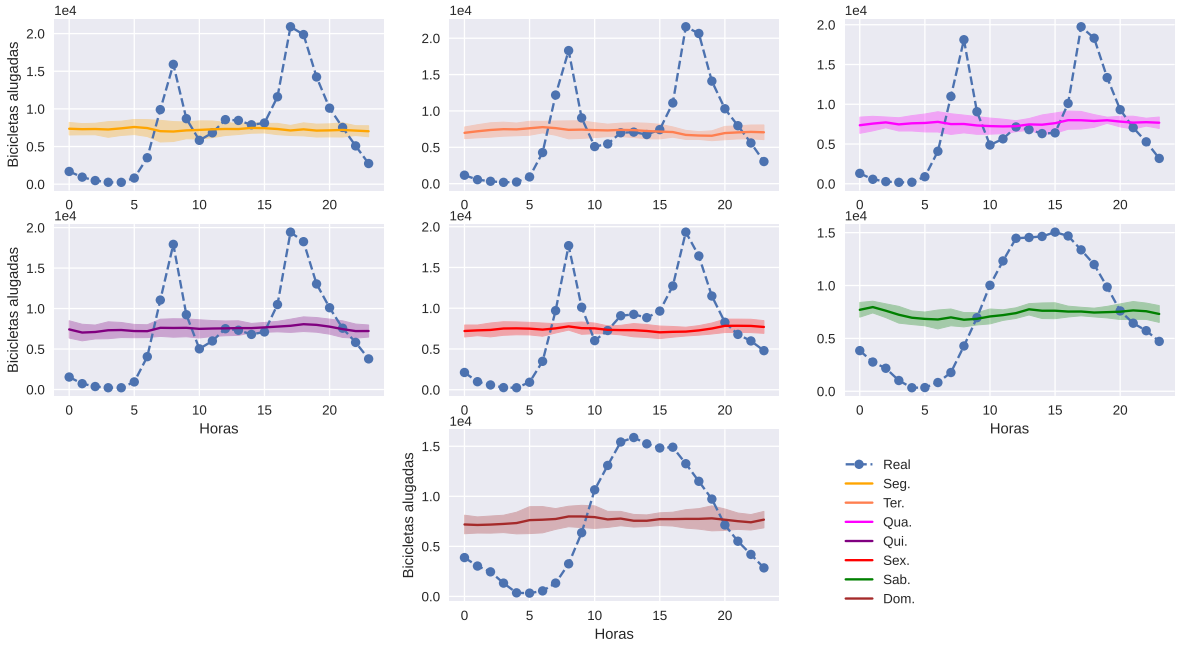


Figura 4.18: Soma por intervalo dos dados sintéticos gerados pelo ARIMA

```
df_sample = p.get_df(lista_datas[:365*24], treino_s,
                     timesteps=24)
dict_weeks_real = {wk:p.get_count(df_sample,wk,timestep=24,
                     column='ts')['cnt_0'] for wk in range(7)}
path_lista_fakes = glob.glob("datasets/generated/timegan/3/*.npy")
dict_weeks_fakes = {i:p.get_list_wks(path_lista_fakes,
                     lista_datas[:365*24], 24, wk=i) for i in range(7)}
p.plot_compare_sum(dict_weeks_fakes,dict_weeks_real,
                     bbox=(1.45, -0.1), figtitle="")
```

O maior desafio da GAN, contudo, é capturar as características dos finais de semana, já que em um ano existem mais dias úteis. Isso faz com que o modelo acabe extrapolando as características dos dias úteis (picos, principalmente) aos finais de semana. Uma alternativa para minimizar esse desafio seria utilizar o parâmetro para o `batch_size` que representa-se apenas alguns dias, ao invés do mês inteiro. Entretanto, isso iria impactar o resultado durante os dias úteis. Outra alternativa, seria fazer modelos específicos para os dias úteis para os finais de semana, com o risco de algumas dependências temporais serem perdidas e com uma complexidade maior para treinamento e

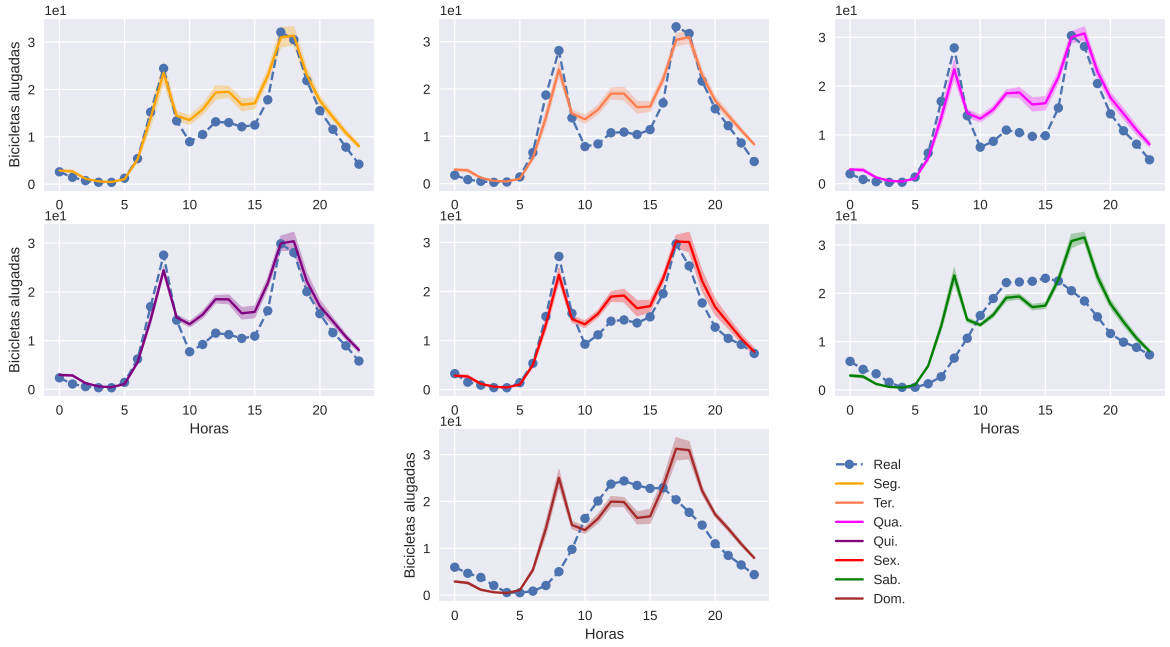


Figura 4.19: Soma por intervalo dos dados sintéticos gerados pela GAN.

avaliação, já que agora seriam necessários 2 modelos.

4.9. Considerações Finais

Este minicurso introduziu o conceito de GANs para a geração de séries temporais. Inicialmente apresentamos alguns exemplos de como as GANs têm sido usado com eficiência em aplicações de séries temporais. Em seguida, fazemos uma breve introdução aos conceitos de séries temporais, suas propriedades e técnicas estatísticas que auxiliam seu entendimento e modelagem. Apresentamos também os conceitos de aprendizado profundo, que são fundamentais ao entendimento da arquitetura das GANs. Introduzimos também o conceito de GANs, seu funcionamento e alguns exemplos de casos de uso da arquitetura original. Finalmente, a partir do que foi exposto ao longo do capítulo, apresentamos os principais desafios de pesquisa de GANs em séries temporais e finalizamos o capítulo um caso de uso prático dos conceitos aprendidos.

A arquitetura GAN é uma tecnologia relativamente nova, mas que usa conceitos bastante conhecidos de aprendizado profundo. Isso faz com que as GANs sejam uma técnica extremamente versátil em aprender e reproduzir as características principais de

diferentes conjuntos de dados, sendo um dos modelos generativos mais promissores, principalmente no campo de visão computacional.

O conceito de séries temporais é utilizado em vários aspectos da vida humana, desde músicas à área da saúde. Nesse contexto, a literatura evidencia que as GANs podem ser utilizadas com eficiência para tratar de aplicações envolvendo séries temporais. Da mesma forma, os desafios e limitações dos exemplos encontrados na literatura mostram que ainda há muito o que ser explorado em relação utilização de GANs para tratar de problemas envolvendo séries temporais.

Agradecimentos

O presente trabalho foi realizado no Laboratório de Pesquisa em Redes e Multimídia (LPRM) do Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal do Espírito Santo (UFES), com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001, do CNPq, da Fundação de Amparo à Pesquisa do Espírito Santo (FAPES) e FAPESP (Grant #2020/05182-3).

Referências

AGGARWAL, A.; MITTAL, M.; BATTINENI, G. Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, Elsevier, v. 1, n. 1, p. 100004, 2021.

ALZANTOT, M.; CHAKRABORTY, S.; SRIVASTAVA, M. Sensegen: A deep learning architecture for synthetic sensor data generation. In: IEEE. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. [S.l.], 2017. p. 188–193.

AMIRIAN, J.; HAYET, J.-B.; PETTRÉ, J. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2019. p. 0–0.

BAGNALL, A. et al. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, Springer, v. 31, n. 3, p. 606–660, 2017.

BEERAM, S. R.; KUCHIBHOTLA, S. Time series analysis on univariate and multivariate variables: a comprehensive survey. *Communication Software and Networks*, Springer, p. 119–126, 2021.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, IEEE, v. 5, n. 2, p. 157–166, 1994.

BOX, G. E. et al. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2015.

BRASIL. *Lei Geral de Proteção de Dados (LGPD)*. 2018. <https://www2.camara.leg.br/legin/fed/lei/2018/lei-13709-14-agosto-2018-787077-publicacaooriginal-156212-pl.html>. Acessado em: 13 Ago. 2021.

BROCK, A.; DONAHUE, J.; SIMONYAN, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

BROCKWELL, P. J.; DAVIS, R. A. *Time series: theory and methods*. [S.l.]: Springer Science & Business Media, 2009.

CARLINI, N. et al. The secret sharer: Evaluating and testing unintended memorization in neural networks. In: *28th USENIX Security Symposium (USENIX Security 19)*. [S.l.: s.n.], 2019. p. 267–284.

CHATFIELD, C. *The analysis of time series: an introduction*. [S.l.]: Chapman and Hall/CRC, 2003.

CHEN, D. et al. Gan-leaks: A taxonomy of membership inference attacks against generative models. In: *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*. [S.l.: s.n.], 2020. p. 343–362.

CHEN, J. et al. Generative dynamic link prediction. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, AIP Publishing LLC, v. 29, n. 12, p. 123111, 2019.

CHOLLET, F. *Deep learning with Python*. [S.l.]: Simon and Schuster, 2021.

COSTA, P. et al. End-to-end adversarial retinal image synthesis. *IEEE transactions on medical imaging*, IEEE, v. 37, n. 3, p. 781–791, 2017.

DAI, W. et al. Scan: Structure correcting adversarial network for organ segmentation in chest x-rays. In: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. [S.l.]: Springer, 2018. p. 263–273.

DONAHUE, C.; MCAULEY, J.; PUCKETTE, M. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

DONG HAO-WEN E HSIAO, W.-Y.; YANG, L.-C. e; YANG, Y.-H. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2018. v. 32, n. 1.

DOUZAL-CHOUAKRIA, A.; AMBLARD, C. Classification trees for time series. *Pattern Recognition*, Elsevier, v. 45, n. 3, p. 1076–1091, 2012.

DWORK, C. Differential privacy: A survey of results. In: SPRINGER. *International conference on theory and applications of models of computation*. [S.l.], 2008. p. 1–19.

ENGEL, J. et al. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

ESTEBAN, C.; HYLAND, S. L.; RÄTSCH, G. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

FRIGERIO, L. et al. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In: SPRINGER. *IFIP International Conference on ICT Systems Security and Privacy Protection*. [S.l.], 2019. p. 151–164.

GAO, C. et al. Adversarialnas: Adversarial neural architecture search for gans. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 5680–5689.

GHOSH, S. et al. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*, 2016.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.

GOODFELLOW, I. et al. Generative adversarial nets. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2014. p. 2672–2680.

GOOGLE. *Web Traffic Timeseries Forecasting*. 2018. <https://www.kaggle.com/c/web-traffic-time-series-forecasting>.

HARTMANN KAY GREGOR E SCHIRRMEISTER, R. T.; BALL, T. Eeg-gan: Generative adversarial networks for electroencephalograhic (eeg) brain signals. *arXiv preprint arXiv:1806.01875*, 2018.

HAYKIN, S. *Neural Networks and Learning Machines*. [S.l.]: Prentice Hall, 2009. (Neural networks and learning machines, v. 10). ISBN 9780131471399.

HOCHREITER, S. et al. *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*. [S.l.]: A field guide to dynamical recurrent neural networks. IEEE Press In, 2001.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HUANG, J.; KURNIAWAN, E.; SUN, S. Cellular kpi anomaly detection with gan and time series decomposition. In: IEEE. *ICC 2022-IEEE International Conference on Communications*. [S.l.], 2022. p. 4074–4079.

JAUHRI, A. et al. Generating realistic ride-hailing datasets using gans. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, ACM New York, NY, USA, v. 6, n. 3, p. 1–14, 2020.

KADRI, F. et al. Towards accurate prediction of patient length of stay at emergency department: a gan-driven deep learning framework. *Journal of Ambient Intelligence and Humanized Computing*, Springer, p. 1–15, 2022.

KAMPOURAKI, A.; MANIS, G.; NIKOU, C. Heartbeat time series classification with support vector machines. *IEEE transactions on information technology in biomedicine*, IEEE, v. 13, n. 4, p. 512–518, 2008.

KARIM, F. et al. Multivariate lstm-fcns for time series classification. *Neural Networks*, Elsevier, v. 116, p. 237–245, 2019.

KARRAS, T. et al. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

KOTSIFAKOS, A.; PAPAPETROU, P. Model-based time series classification. In: SPRINGER. *International Symposium on Intelligent Data Analysis*. [S.l.], 2014. p. 179–191.

LIM, B.; ZOHREN, S. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, The Royal Society Publishing, v. 379, n. 2194, p. 20200209, 2021.

LIN, Z. et al. Using gans for sharing networked time series data: Challenges, initial promise, and open questions. In: *Proceedings of the ACM Internet Measurement Conference*. [S.l.: s.n.], 2020. p. 464–483.

LUO, Y. et al. Multivariate time series imputation with generative adversarial networks. *Advances in neural information processing systems*, v. 31, 2018.

MAHARAJ, E. A.; ALONSO, A. M. Discriminant analysis of multivariate time series: Application to diagnosis based on ecg signals. *Computational Statistics & Data Analysis*, Elsevier, v. 70, p. 67–87, 2014.

Malandrino, F.; Chiasserini, C.; Kirkpatrick, S. Cellular network traces towards 5g: Usage, analysis and generation. *IEEE Transactions on Mobile Computing*, v. 17, n. 3, p. 529–542, 2018.

MOGREN, O. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.

MONTGOMERY, D. C.; HINES, W. W. *Probability and statistics in engineering and management science*. [S.l.]: John Wiley & Sons, 1980.

OH, E. et al. Sting: Self-attention based time-series imputation networks using gan. In: IEEE. *2021 IEEE International Conference on Data Mining (ICDM)*. [S.l.], 2021. p. 1264–1269.

POVINELLI, R. J. et al. Time series classification using gaussian mixture models of reconstructed phase spaces. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 16, n. 6, p. 779–783, 2004.

QU, Y. et al. Gan-driven personalized spatial-temporal private data sharing in cyber-physical social systems. *IEEE Transactions on Network Science and Engineering*, IEEE, 2020.

RAMOS, H. S. et al. Aprendizado federado aplicado à internet das coisas. *Sociedade Brasileira de Computação*, 2021.

RAO, J. et al. Lstm-trajgan: A deep learning approach to trajectory privacy protection. *arXiv preprint arXiv:2006.10521*, 2020.

RIBEIRO, I. et al. Mobility and community detection based on topics of interest. In: *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*. [S.l.]: IEEE, 2021. p. 1–6.

RIBEIRO, I. et al. Uma abordagem para geração de séries temporais de mobilidade urbana baseada em aprendizado profundo. In: *Anais do V Workshop de Computação Urbana*. Porto Alegre, RS, Brasil: SBC, 2021. p. 251–264. ISSN 2595-2706.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *Learning internal representations by error propagation*. [S.l.], 1985.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.

SALIMANS, T. et al. Improved techniques for training gans. *Advances in neural information processing systems*, v. 29, p. 2234–2242, 2016.

SCOTT, J. et al. *CRAWDAD dataset cambridge/haggle (v. 2009-05-29)*. 2009. Downloaded from https://crawdad.org/cambridge/haggle/20090529.

SHUMWAY, R. H.; STOFFER, D. S. *Time series analysis and its applications*. [S.l.]: Springer, 2000. v. 3.

SINGH, H.; RAY, M. R. Synthetic stream flow generation of river gomti using arima model. In: *Advances in Civil Engineering and Infrastructural Development*. [S.l.]: Springer, 2021. p. 255–263.

SINGH, N. K.; RAZA, K. Medical image generation using generative adversarial networks: A review. *Health informatics: A computational perspective in healthcare*, Springer, p. 77–96, 2021.

SUSSKIND, J.; ANDERSON, A.; HINTON, G. E. *The Toronto face dataset*. [S.l.], 2010.

SYKACEK, P.; ROBERTS, S. J. Bayesian time series classification. *Advances in Neural Information Processing Systems*, v. 14, 2001.

WANG, Z. et al. Data-augmentation-based cellular traffic prediction in edge-computing-enabled smart city. *IEEE Transactions on Industrial Informatics*, IEEE, v. 17, n. 6, p. 4179–4187, 2020.

WEI, L.; KEOGH, E. Semi-supervised time series classification. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2006. p. 748–753.

WUNSCH, D.; XU, R. *Clustering*. [S.l.]: John Wiley & Sons, 2008.

YI, X.; WALIA, E.; BABYN, P. Unsupervised and semi-supervised learning with categorical generative adversarial networks assisted by wasserstein distance for dermoscopy image classification. *arXiv preprint arXiv:1804.03700*, 2018.

YI, X.; WALIA, E.; BABYN, P. Generative adversarial network in medical imaging: A review. *Medical image analysis*, Elsevier, v. 58, p. 101552, 2019.

YOON, J.; JARRETT, D.; SCHAAR, M. van der. Time-series generative adversarial networks. In: WALLACH, H. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. v. 32. Disponível em: <https://proceedings.neurips.cc/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf>.

YU, L. et al. Seqgan: Sequence generative adversarial nets with policy gradient. In: *Proceedings of the AAAI conference on artificial intelligence*. [S.l.: s.n.], 2017. v. 31, n. 1.

ZAKI, M. J.; JR, W. M. *Data mining and machine learning: Fundamental concepts and algorithms*. [S.l.]: Cambridge University Press, 2020.

ZHANG, A. et al. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

ZHANG, C. et al. Generative adversarial network for synthetic time series data generation in smart grids. In: IEEE. *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. [S.l.], 2018. p. 1–6.

ZHANG, D.; MA, M.; XIA, L. A comprehensive review on gans for time-series signals. *Neural Computing and Applications*, Springer, p. 1–21, 2022.

ZHANG, G. P. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, Elsevier, v. 50, p. 159–175, 2003.

ZHANG, L. Stggan: Spatial-temporal graph generation. In: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. [S.l.: s.n.], 2019. p. 608–609.

ZHANG, Y. et al. Gcgan: Generative adversarial nets with graph cnn for network-scale traffic prediction. In: IEEE. *2019 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2019. p. 1–8.

ZHANG, Y. et al. Trafficgan: Network-scale deep traffic prediction with generative adversarial nets. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 22, n. 1, p. 219–230, 2019.

ZHENG, Y. et al. Mining interesting locations and travel sequences from gps trajectories. In: ACM. *World wide web*. [S.l.], 2009. p. 791–800.

Índice de Autores

WebMedia2022

Minicursos - XXVIII Simpósio Brasileiro de Sistemas Multimídia e Web

Patrocínio

nic.br globo

CAPES CNPq

cgi.br Localiza labs

Cooperação

acm In-Cooperation

sigweb

SIGMM

Organização

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Realização

SBC
Sociedade Brasileira de Computação