



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

**TALES ARAUJO DOS SANTOS**

**A UTILIZAÇÃO DE SISTEMAS DE RECOMENDAÇÃO NO CONTEXTO DA  
DESINFORMAÇÃO**

**FORTALEZA**

**2023**

TALES ARAUJO DOS SANTOS

A UTILIZAÇÃO DE SISTEMAS DE RECOMENDAÇÃO NO CONTEXTO DA  
DESINFORMAÇÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. José Maria da Silva Monteiro Filho.

Coorientador: Prof. Dr. José Wellington Franco da Silva.

FORTALEZA

2023

TALES ARAUJO DOS SANTOS

A UTILIZAÇÃO DE SISTEMAS DE RECOMENDAÇÃO NO CONTEXTO DA  
DESINFORMAÇÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: 30/06/2023.

BANCA EXAMINADORA

---

Prof. Dr. José Maria da Silva Monteiro  
Filho (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. José Wellington Franco da  
Silva (Coorientador)  
Universidade Federal do Ceará (UFC)

---

Prof(a). Amanda Drielly Pires Venceslau  
Universidade Federal do Ceará (UFC)

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

## AGRADECIMENTOS

Agradeço à minha mãe, Gilma de Araujo, pelo enorme amor, mansidão e carinho, por me colocar no caminho da educação e fazer todo sacrifício possível (e impossível) para que eu pudesse estar onde estou hoje.

Ao meu pai, Valdemir José, pela infinita paciência, gentileza e perseverança, por me incentivar a trilhar o caminho da educação e lutar com o mundo para proteger a nossa família. Sem você eu também não teria chegado aqui.

Ao meu irmão, Tierre Araujo, por ser sempre uma pessoa incrível, empática e ser meu grande amigo e companheiro nas horas difíceis. Eu também não teria suportado o caminho árduo que foi chegar até aqui sem você.

À Profa. Dra. Lígia Barros Costa, por ser meu grande norte e referência em educação desde a infância, por me auxiliar e me incentivar a buscar o caminho do conhecimento e prestar todo o suporte necessário para que eu pudesse trilhar o caminho que me trouxe aqui.

Ao Prof. Dr. José Maria da Silva Monteiro Filho, por prestar um enorme suporte no meu crescimento acadêmico, pela orientação deste trabalho e por conceder todo o auxílio necessário para que este pudesse ser conduzido com plenitude.

Ao Prof. Dr. José Wellington Franco da Silva, pela orientação deste trabalho, pelos grandes ensinamentos na área de Sistemas de Recomendação e por apresentar ótimos *insights* a respeito do desenvolvimento do meu trabalho de conclusão de curso.

À Profa. Amanda Drielly Pires Venceslau por fazer parte da banca avaliadora e demonstrar interesse em meu trabalho. Muito obrigado.

Aos meus amigos de infância, Márcio e Ailton, pela amizade até os dias de hoje, pessoas na qual tenho muita estima e guardo grande apreço no coração por me incentivarem a traçar o caminho que estou trilhando hoje.

Aos meus amigos da época do colégio, em especial Ricardo e Felipe, por serem pessoas incríveis e extremamente empáticas que conheci durante esta jornada, pois lutaram junto e me apoiaram em tudo que pude pleitear desde os dias de hoje. Vou sentir saudades dos grandes momentos que tivemos na universidade.

Aos meus amigos do LEAD, Artur, Gerônimo e demais, cuja amizade e troca de experiência forneceram insumos essenciais para inspirar o aprofundamento na área de Ciência de Dados e a me tornar uma pessoa melhor. Agradeço ao Artur por me ajudar na revisão deste trabalho.

A todos os meus professores que me apoiaram com tudo que puderam, em meu tempo de escola, para que eu trilhasse e não desistisse do caminho da educação. Ao Professor Marcos, Professora Tatiane, Professor Kilder e todos os demais presentes em meu caminho como discente. Devo a todos vocês minha mais profunda gratidão.

Aos colegas da turma de graduação, pelas reflexões, críticas e sugestões recebidas.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do template utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

Por fim, agradeço a todos aqueles que participaram direta e indiretamente de toda minha graduação. Obrigado.

"Quem, de três milênios, não é capaz de se dar conta, vive na ignorância, na sombra, à mercê dos dias, do tempo."

(Johann Goethe)

## RESUMO

A “pandemia” da desinformação, facilitada pelas redes sociais e plataformas de compartilhamento de mensagens móveis, tem deixado de ser um simples incômodo para impactar severamente a lei e a ordem por meio da manipulação deliberada e em larga escala do sentimentos do público em geral. Os sistemas de recomendação são um importante conjunto de técnicas e de ferramentas de software capazes de atuar no problema de filtragem de conteúdo e de recuperação de informação num cenário esmagador de possibilidades de escolhas, que apresenta o mundo tecnológico nos dias atuais. Uma dessas plataformas onde ocorre grande fluxo de dados no cenário brasileiro é o aplicativo WhatsApp, que tem sido alvo de agentes compartilhadores de conteúdo desinformativo. Com o presente trabalho, pretendemos conduzir um experimento em uma tentativa de descobrir como sistemas de recomendação poderiam atuar para recomendar mensagens de caráter semelhantes, no que concerne ao seu conteúdo factual. Iremos fazer uma demonstração de como as técnicas empregadas poderiam representar um sistema de redomendação, apresentando suas arquiteturas, métodos de similaridade, listas de recomendações, bem como seus mecanismos de avaliação de performance. Apresentaremos como os dados em estudo, relativos a mensagens coletadas do WhatsApp que concernem à pandemia de COVID-19, se comportaram com a arquitetura proposta, bem como seus resultados em relação ao desempenho das recomendações, de forma a entender métricas de performance como o *mean reciprocal rank* e o *mean average precision* caracterizam a relevância trazida pelos resultados ao se utilizar um critério simples de semelhança - a similaridade do cosseno - em conteúdos puramente textuais. Por fim, avaliaremos a capacidade do próprio sistema de recomendação de ser utilizado como parte de um classificador para mensagens que não possuem rótulo de presença de desinformação atribuído, uma vez que o classificador utiliza um critério ingênuo para identificar a presença (ou não) de conteúdo desinformativo.

**Palavras-chave:** Sistemas de Recomendação; Desinformação; Similaridade de Textos; WhatsApp; COVID-19.

## ABSTRACT

The misinformation “pandemic”, facilitated by social media and mobile messaging platforms, has gone from being a minor nuisance to severely impacting law and order through the deliberate and large-scale manipulation of the general public’s sentiments. Recommender systems are an essential set of techniques and software tools capable of acting on the problem of content filtering and information retrieval in an overwhelming scenario of choice possibilities that presents the technological world today. One platform with a large data flow in the Brazilian scenario is the WhatsApp application, which has been the target of agents sharing disinformation content. With the present work, we intend to experiment to discover how recommender systems could act to recommend messages of a similar character concerning their factual content. We will demonstrate how the employed techniques could represent a recommender system, presenting their architectures, similarity methods, lists of recommendations and performance evaluation mechanisms. We will also present how the data under study, regarding messages collected from WhatsApp concerning the COVID-19 pandemic, behaved with the proposed architecture, as well as their results regarding the performance of recommendations, in order to understand how performance metrics such as mean reciprocal rank and mean average precision characterize the relevance brought by the results when using a simple criterion of similarity - cosine similarity - in purely textual contents. Finally, the ability of the recommendation system itself to be used as part of a classifier for messages that do not have a assigned label of disinformation presence will be evaluated, since the classifier uses a naive approach to identify the presence (or absence) of disinformation content.

**Keywords:** Recommendation Systems; Misinformation; Text Similarity; WhatsApp; COVID-19.

## LISTA DE FIGURAS

Figura 1 – Disposição dos tipos de informação de má qualidade . . . . .	20
Figura 2 – Relações entre os conceitos associados à desinformação . . . . .	22
Figura 3 – Mecanismo de <i>eco</i> da desinformação . . . . .	23
Figura 4 – Representação intuitiva do formato <i>bag of words</i> . . . . .	24
Figura 5 – Etapas da filtragem colaborativa . . . . .	30
Figura 6 – Visão geral do paradigma orientado a conteúdo . . . . .	33
Figura 7 – Motor de busca hipotético . . . . .	41
Figura 8 – Definição de condições sobre a busca . . . . .	42
Figura 9 – Cálculo do Recall@K . . . . .	44
Figura 10 – Execução das buscas para o cálculo de MRR . . . . .	45
Figura 11 – Diretrizes de rotulação das mensagens . . . . .	56
Figura 12 – Síntese do processo de geração de dados . . . . .	56
Figura 13 – Visão do conjunto de dados . . . . .	57
Figura 14 – Visualização de dados faltantes . . . . .	58
Figura 15 – Distribuição das classes . . . . .	59
Figura 16 – Distribuição da contagem de sentenças . . . . .	61
Figura 17 – Distribuição da contagem de palavras com caixa alta . . . . .	62
Figura 18 – Exemplos de mensagens desinformativas escolhidas aleatoriamente . . . . .	63
Figura 19 – Exemplos de mensagens não-desinformativas escolhidas aleatoriamente . . . . .	64
Figura 20 – Nuvem de palavras da desinformação . . . . .	65
Figura 21 – Nuvem de palavras da informação . . . . .	65
Figura 22 – Visão de Alto Nível do Sistema de Recomendação . . . . .	68
Figura 23 – Divisão dos dados . . . . .	73
Figura 24 – Mensagem aleatória selecionada do corpus de treino . . . . .	74
Figura 25 – Lista de recomendações para a mensagem da Figura 24 . . . . .	74
Figura 26 – Mensagem aleatória selecionada do corpus de treino . . . . .	75
Figura 27 – Lista de recomendações para a mensagem da Figura 26 . . . . .	75
Figura 28 – Mensagem aleatória sem rótulo (mensagem real) . . . . .	75
Figura 29 – Lista de recomendações para a mensagem da Figura 28 . . . . .	76
Figura 30 – Amostragem de Consultas ao SR . . . . .	78
Figura 31 – Critério de Classificação . . . . .	80

Figura 32 – Matriz de Confusão . . . . . 81

## LISTA DE TABELAS

Tabela 1 – Busca #1 . . . . .	46
Tabela 2 – Busca #2 . . . . .	46
Tabela 3 – Busca #3 . . . . .	47
Tabela 4 – Contagem de sentenças . . . . .	62
Tabela 5 – Contagem de palavras com caixa alta . . . . .	62
Tabela 6 – Avaliação da performance . . . . .	78
Tabela 7 – Desempenho do classificador . . . . .	81

## LISTA DE ABREVIATURAS E SIGLAS

AP	<i>Average Precision</i> / Precisão Média
API	<i>Application Programming Interface</i> / Interface de Programação de Aplicação
FANAR	<i>Fake News Aware Recommender System</i> / Sistema de Recomendação Sensível a Notícias Falsas
FNEWR	<i>Fake News-Aware Recommendation Dataset</i> / Conjunto de Dados de Recomendação Sensível a Notícias Falsas
MAP	<i>Mean Average Precision</i> / Média das Precisas Médias
MRR	<i>Mean Reciprocal Rank</i> / Classificação de Significância Recíproca
PLN	<i>Natural Language Processing</i> / Processamento de Linguagem Natural
SMOTE	<i>Synthetic Minority Oversampling Technique</i> / Técnica de Sobreamostragem Minoritária Sintética
SR	<i>Recommendation System</i> / Sistema de Recomendação
TF-IDF	<i>Term Frequency – Inverse Document Frequency</i> / Frequência do Termo - Frequência Inversa dos Documentos

## LISTA DE SÍMBOLOS

$d$	Distância Euclidiana
$ES$	Similaridade Euclidiana
$IDF$	Frequência Inversa dos Documentos
$J$	Índice de Jaccard
$TF$	Frequência de Termos
$r$	Coefficiente de Correlação Amostral
$u$	Função Utilidade
$\mathbb{E}$	Valor Esperado
$\rho$	Coefficiente de Correlação de Pearson

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	16
<b>1.1</b>	<b>Disposição dos Elementos do Trabalho</b>	17
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	19
<b>2.1</b>	<b>Desinformação</b>	19
<i>2.1.1</i>	<i>Conceitos Associados</i>	20
<i>2.1.2</i>	<i>Mecanismos da Desinformação</i>	22
<b>2.2</b>	<b>Modelagem Semântico-Distribucional</b>	23
<i>2.2.1</i>	<i>Bag of Words</i>	24
<i>2.2.2</i>	<i>TF-IDF</i>	25
<b>2.3</b>	<b>Sistemas de Recomendação</b>	26
<i>2.3.1</i>	<i>Formalização do Problema</i>	27
<i>2.3.2</i>	<i>Modelagens e Técnicas de Recomendação</i>	28
<i>2.3.2.1</i>	<i>Filtragem Colaborativa</i>	29
<i>2.3.2.2</i>	<i>Filtragem Baseada em Conteúdo</i>	32
<i>2.3.2.3</i>	<i>Modelos Híbridos</i>	35
<b>2.4</b>	<b>Funções de Similaridade</b>	36
<i>2.4.1</i>	<i>Distância Euclidiana</i>	37
<i>2.4.2</i>	<i>Similaridade do Cosseno</i>	37
<i>2.4.3</i>	<i>Correlação de Pearson</i>	38
<i>2.4.4</i>	<i>Índice de Jaccard</i>	39
<b>2.5</b>	<b>Métricas de Avaliação de Desempenho</b>	40
<i>2.5.1</i>	<i>Métricas Sensíveis à Ordem</i>	43
<i>2.5.1.1</i>	<i>Recall@K</i>	43
<i>2.5.2</i>	<i>Métricas Insensíveis à Ordem</i>	43
<i>2.5.2.1</i>	<i>Mean Reciprocal Rank</i>	43
<i>2.5.2.2</i>	<i>Mean Average Precision@k ou Média das Precisoões Médias em k</i>	45
<b>2.6</b>	<b>Considerações Finais do Capítulo</b>	47
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	49
<b>3.1</b>	<b>Contextualização</b>	49
<b>3.2</b>	<b>Considerações Finais do Capítulo</b>	51

4	<b>FAKEWHATSAPP.BR</b> . . . . .	52
4.1	<b>Contexto e Razões</b> . . . . .	52
4.2	<b>Estratégia de Criação do <i>Corpus</i></b> . . . . .	53
4.2.1	<i>Coleta de Dados Brutos</i> . . . . .	53
4.2.2	<i>Normalização dos Dados</i> . . . . .	54
4.2.3	<i>Rotulação das Classes</i> . . . . .	55
4.3	<b>Análise Exploratória</b> . . . . .	57
4.3.1	<i>Extração de Informação</i> . . . . .	58
4.3.2	<i>Visualização de Frequência de palavras</i> . . . . .	60
5	<b>O SISTEMA DE RECOMENDAÇÃO PROPOSTO</b> . . . . .	66
5.1	<b>Contexto e Iniciativa</b> . . . . .	66
5.2	<b>Arquitetura do SR</b> . . . . .	67
5.3	<b>Implementação</b> . . . . .	68
5.3.1	<i>Pré-processamento</i> . . . . .	68
5.3.2	<i>Vetorização</i> . . . . .	69
5.3.3	<i>Cálculo da Similaridade</i> . . . . .	69
5.3.4	<i>Função de Recomendação</i> . . . . .	70
6	<b>AVALIAÇÃO EXPERIMENTAL</b> . . . . .	72
6.1	<b>Experimentos</b> . . . . .	72
6.2	<i>Running Examples</i> . . . . .	73
6.3	<b>Resultados</b> . . . . .	76
6.3.1	<i>Avaliação da Performance do SR</i> . . . . .	77
6.3.2	<i>Avaliação do SR como Sistema Rotulador</i> . . . . .	79
7	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . .	83
7.1	<b>Trabalhos Futuros</b> . . . . .	84
	<b>REFERÊNCIAS</b> . . . . .	86

## 1 INTRODUÇÃO

Nos últimos anos, a *epidemia da informação*, isto é, a informação que se espalha de maneira deliberada sem moderação de autenticidade factual conforme prevê Zarocostas (2020), tem impactado gravemente a manutenção das políticas de saúde pública, de desenvolvimento econômico e infraestrutura mundiais. Especialmente, o contexto evidenciado pelo autor anterior destaca o impacto da desinformação no que concerne à pandemia causada pelo vírus SARS-CoV-2, causador da COVID-19, originada em Dezembro de 2019 em Wuhan, China. Essa “*infodemia*” tem como mecanismo principal de atuação a desinformação que, segundo Woolley e Howard (2016), é aquela que é incorreta e tem o propósito malicioso de provocar e incitar pessoas negativamente.

Existem diversos mecanismos atrelados à desinformação que trabalham diferentes formas de propagação e essas formas podem incorporar ambientes propícios ao desenvolvimento e organização de núcleos de compartilhamento de conteúdos chamados *virais*, aqueles que se espalham rapidamente e geram alto engajamento (CANAVILHAS *et al.*, 2019). Particularmente, no cenário brasileiro, existe a plataforma de comunicação de mensagens instantâneas WhatsApp<sup>1</sup>, extremamente popular no país, utilizada quase que como artefato obrigatório de comunicação social via internet. Um dos propósitos da plataforma é realizar o envio de mensagens, de diversos tipos de mídia (texto, imagem, áudio) de maneira privada, através de seus *smartphones*. Essa plataforma possui a capacidade de criação de *grupos*, locais onde usuários selecionados podem trocar mensagens entre si. Especialmente esses grupos têm sido alvos de compartilhamento de notícias falsas, conforme aponta Newman *et al.* (2021), e mesmo que algumas dessas informações possam ter sido provadas como falsas, elas continuam a ser replicadas (RESENDE *et al.*, 2019).

De acordo com Ricci *et al.* (2022), os Sistemas de Recomendação (SR) são uma importante subclasse de Sistemas de Filtragem de Informação que fornecem sugestões para itens que são mais relevantes para um usuário em particular. Grbovic (2023) afirma que as empresas de comércio e varejo eletrônico, por exemplo, utilizam o poder da recomendação articulada para alavancar o número de vendas e aumentar o valor entregue aos seus usuários reduzindo o conjunto de prováveis escolhas baseado em critérios de relevância. Dessa forma, eles são ferramentas poderosas capazes de filtrar apenas as informações que atendem a uma certa característica desejada, o que pode sugerir a possibilidade de análise acerca de certos conteúdos informativos com essa ideia em foco.

---

<sup>1</sup> <https://www.whatsapp.com>

Com o presente trabalho, pretende-se realizar o processo de análise de dados extraídos e processados do aplicativo WhatsApp, que contemplam especificamente o contexto da pandemia de COVID-19 no Brasil, para que se possa sustentar a elaboração de um sistema de recomendação que possa utilizar essas mensagens coletadas, que estão definidas de acordo com uma série de critérios que caracterizam desinformação, de forma que esse sistema possa sugerir mensagens com características similares e de mesmo caráter informativo (ou desinformativo). Desta forma, este trabalho poderá ser utilizado como ponto de partida para a construção de outros sistemas de caráter similar que possam realizar a filtragem de conteúdo de diversas fontes de mídia, como sites agregadores de notícias, por exemplo.

Um procedimento detalhado da análise da performance deste sistema de recomendação será realizado, em conjunto com um processo de avaliação da sua capacidade de filtragem, tendo em vista os critérios estabelecidos na etapa de análise exploratória que possam eventualmente caracterizar (ou não) desinformação. Por fim, serão utilizadas métricas de avaliação de desempenho recomendadas para sistemas de recomendação de forma a verificar a capacidade do sistema e sua arquitetura proposta de retornarem resultados relevantes e esperados aos usuários, de forma a avaliar diferentes tamanhos de listas de recomendação.

## **1.1 Disposição dos Elementos do Trabalho**

Este trabalho está organizado em sete capítulos, que são descritos de maneira breve a seguir:

### **– Capítulo 1. Introdução**

O presente capítulo, que aborda o problema, as justificativas, o objetivo do trabalho e a importância de desenvolvimento do mesmo;

### **– Capítulo 2. Referencial Teórico**

Apresentaremos toda a base teórica necessária para o entendimento dos termos, técnicas e metodologias utilizadas pelo desenvolvimento do sistema de recomendação. Serão abordados a fundo os conceitos de desinformação, seus mecanismos de disseminação, bem como a apresentação de uma das bases de Processamento de Linguagem Natural (PLN), a vetorização, que nos tornará capazes de transformar o texto humano para algo concebível para o entendimento dos algoritmos computacionais, e por fim, iremos discorrer sobre os mais importantes sistemas recomendadores atualmente disponíveis, ressaltando suas vantagens e desvantagens, além de elencar formas de avaliar o desempenho e a

performance dos mesmos de acordo com o critério de utilidade.

– **Capítulo 3. Trabalhos Relacionados**

Este capítulo traz uma revisão bibliográfica descritiva de abordagens e resultados alcançados na literatura acerca de problemas similares aos enfrentados neste trabalho.

– **Capítulo 4. O conjunto de dados FakeWhatsApp.BR**

Aqui apresentaremos os detalhes de coleta de informação, processamento e filtragem de dados, bem como a execução da análise exploratória de características pertinentes às mensagens concebidas a partir do WhatsApp. Serão elencados aspectos que podem sugerir a tendência que caracteriza conteúdos desinformativos a partir de análises textuais e estatísticas.

– **Capítulo 5. O SR proposto**

Neste capítulo apresentaremos os detalhes de construção e implementação do sistema de recomendação proposto para este trabalho, de forma a apresentar suas peculiaridades, características e limitações.

– **Capítulo 6. Avaliação Experimental**

Aqui realizaremos procedimentos de demonstração e atuação do sistema de recomendação desenvolvido de forma a avaliar seu funcionamento. Iremos executar procedimentos de aferição de desempenho para extrair uma hipótese conclusiva acerca do quão bem o sistema de recomendação está atuando de acordo com uma série de critérios estabelecidos por algumas métricas de avaliação, que serão discutidas no Capítulo 2.

– **Capítulo 7. Conclusões e Trabalhos Futuros**

Nesta etapa, discorreremos sobre os pontos importantes de elaboração do trabalho que puderam servir de base para a construção do sistema de recomendação em questão, discutir suas vantagens e limitações, de acordo com as análises fornecidas pelos procedimentos do Capítulo 6, de forma a propor novas soluções e desdobramentos de melhorias acerca do que já foi desenvolvido.

## 2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os conceitos fundamentais abordados neste trabalho. Inicialmente, discorreremos sobre a definição de desinformação e conceitos relacionados, além das características de desinformação nas redes sociais. Em seguida, são discutidas as definições de sistemas de recomendação, suas problemáticas e desafios, apresentaremos as definições lógico-matemáticas de cálculo de similaridade entre itens num contexto recomendativo e em seguida iremos propor métodos que podem ser utilizados para o passo de avaliação do desempenho do sistema de recomendação.

### 2.1 Desinformação

De acordo com Floridi (2002), a *informação* é a percepção abstrata do conhecimento que é expressa diretamente sob a forma escrita, oral ou audiovisual. No seu nível mais fundamental, ela diz respeito à *interpretação* daquilo que pode ser *sentido*. Ela não é o conhecimento em si, mas o *significado* que pode ser derivado a partir de sua representação por meio de interpretação, conforme afirma Yockey (2005). A informação está sob a influência, portanto, do conceito de *perspectiva*, desta forma, seu significado pode alterar (seja de maneira qualitativa ou quantitativa) o conhecimento pré-existente do sistema (humano, animal ou máquina) que a recebe, de acordo com seu referencial e estado anterior.

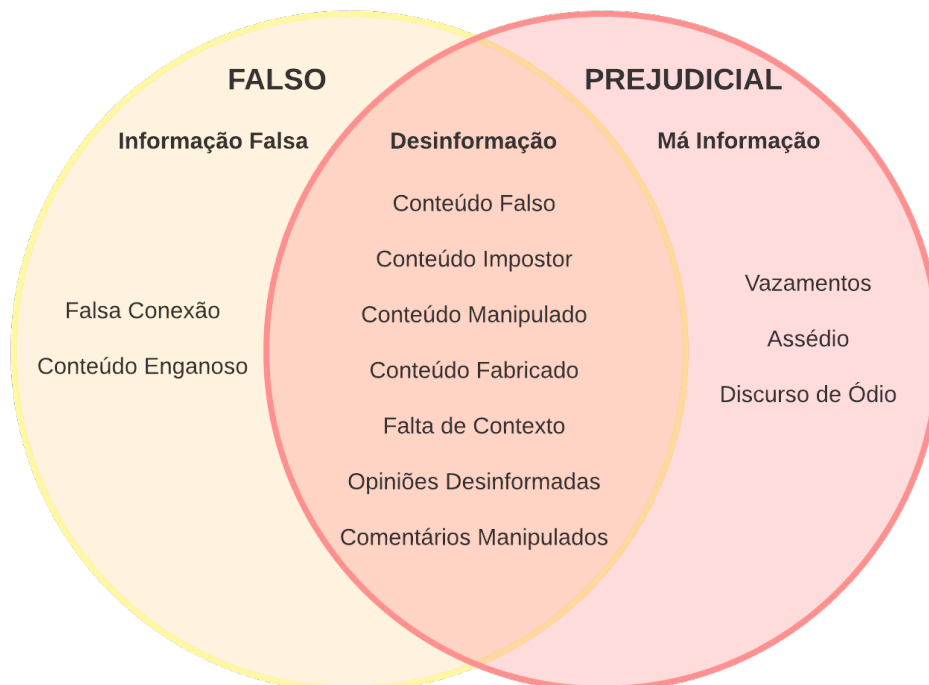
Uma vez que a informação está sujeita à influência da perspectiva e da subjetividade, ela poderá ter a sua forma material alterada com pouco esforço, e isto abre margem para que esta interpretação do conhecimento possa ir de encontro de maneira conflitante com a *veracidade* dos fatos. A língua inglesa define o termo *misinformation*, em tradução livre, “*má informação*”, aquela informação que tem contradição com o fato real, ou seja, é simplesmente incorreta por não se verificar válida na lógica inquestionável. Esta definição está assim proposta para que se possa fazer a diferença ao termo *disinformation*, em tradução direta, o tema deste tópico, **desinformação**. Segundo Woolley e Howard (2016), esta última é deliberadamente falsa, ou seja, possui um propósito atrelado de intencionalmente enganar pessoas.

Nos últimos anos, a inovação tecnológica, sob o aspecto da melhoria dos mecanismos de infraestrutura e capacidade computacional, bem como o avanço da inclusão digital permitiram que grandes plataformas de comunicação pudessem se estabelecer. A ascensão das redes sociais inaugurou uma nova era de conectividade e compartilhamento de informação sem precedentes.

No entanto, também permitiu que a desinformação e as chamadas campanhas de notícias falsas proliferassem e florescessem. A informação *digital*, como as que ocorrem nas redes sociais através da internet, pode influenciar a opinião pública, criar desconfiança e divisão e impactar a tomada de decisões e criação de políticas (JING F. LI; CHOO, 2022). Dado o panorama definido por esta problemática, detectar desinformação nas redes sociais se tornou um problema *extremamente* importante.

Podemos visualmente enumerar exemplos das diferentes nuances de informação com qualidade questionável como demonstrado no diagrama abaixo (Figura 1).

Figura 1 – Disposição dos tipos de informação de má qualidade



Fonte: o autor.

Apesar de fazermos esta divisão clara acerca das diferentes nuances de desinformação definidas pela literatura, consideraremos doravante o termo em português *desinformação* fazendo referência aos conceitos relacionados, aos semelhantes e àqueles sob o mesmo leque da mesma, uma vez que algumas destas nuances serão evidenciadas e exploradas no presente trabalho.

### 2.1.1 *Conceitos Associados*

De acordo com Egelhofer e Lecheler (2019), é evidente que a desinformação no discurso político não é algo recente, entretanto, a extensão na qual ela ocorre nos dias atuais aparenta estar em crescimento. A seguir, faremos algumas definições buscando uma ampla

contextualização dentro da literatura de comunicação política existente, pois desta forma, demonstraremos que, apesar de não serem conceitos novos, são sintomas relacionados que reforçam este crescimento de longo período da desinformação.

Segundo O'Donnell e Jowett (2004), **propaganda** descreve uma classe específica e abrangente de comunicação que pode ser descrita como a tentativa deliberada e sistemática de moldar percepções, manipular pensamentos e direcionar o comportamento para alcançar uma resposta que promova a intenção desejada do propagandista. Ao contrário da prática da persuasão, o verdadeiro propósito da propaganda permanece oculto. Para atingir esse objetivo, os propagandistas visam controlar o fluxo de informações, muitas vezes apresentando informações distorcidas do que parece ser uma fonte confiável.

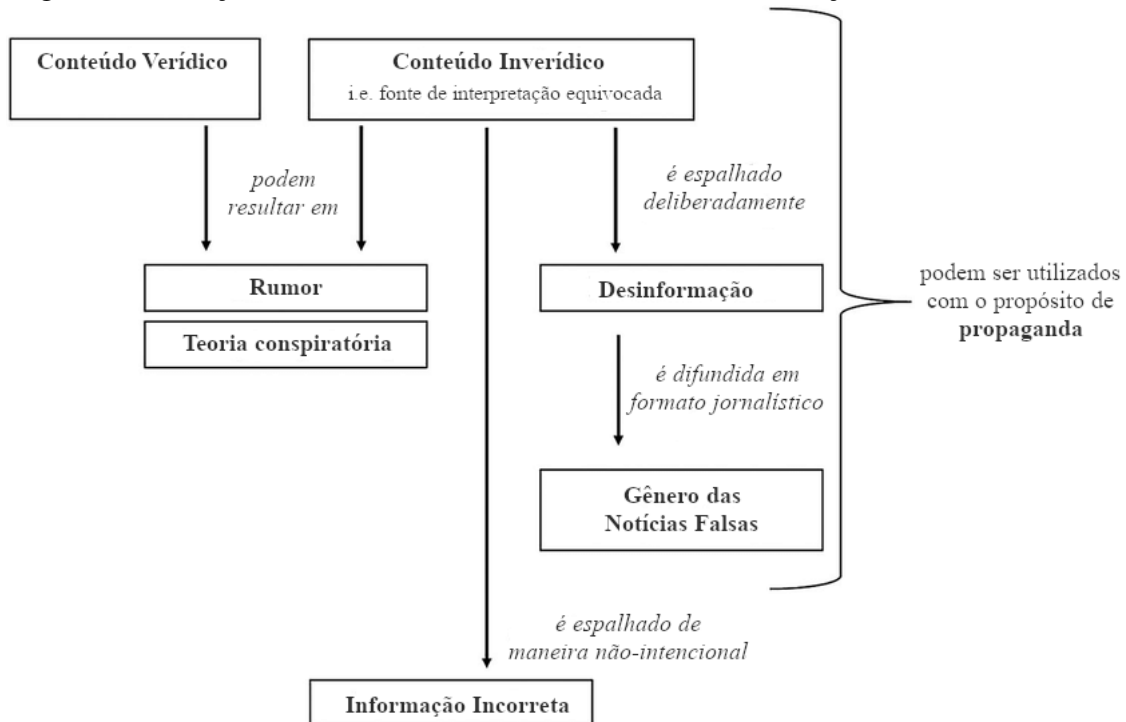
As **notícias falsas**, que se tornaram populares sob a denominação em inglês *fake news*, são informações falsas ou enganosas apresentadas como notícias. Elas geralmente têm o objetivo de prejudicar a reputação de uma pessoa ou entidade ou ganhar dinheiro com receita de publicidade (HUNT, 2016; SCHLESINGER, 2017). Embora as notícias falsas sempre tenham sido espalhadas ao longo da história, o termo foi usado pela primeira vez na década de 1890, quando reportagens sensacionalistas nos jornais eram comuns (MERRIAM-WEBSTER, 2017; SOLL, 2016). No entanto, o termo não tem uma definição fixa e tem sido amplamente aplicado a qualquer tipo de informação falsa. Também tem sido usado por pessoas (ou grupos) de perfil social altamente engajador para aplicar a qualquer notícia desfavorável a eles.

**Rumores e teorias da conspiração** são outros dois conceitos recorrentemente mencionados no contexto de desinformação. De acordo com Berinsky (2017), os *rumores* são caracterizados principalmente por sua falta de evidências e na sua transmissão social generalizada. As *teorias da conspiração* podem ser distinguidas dos rumores por seu esforço para explicar algum evento ou prática ao fazer referência a supostos esquemas obscuros de pessoas poderosas, que tentam ocultar seus papéis (VERMEULE; SUNSTEIN, 2009). Por meio de argumentos excessivamente simplistas, as teorias conspiratórias fazem as pessoas terem entendimento de questões complexas ao oferecerem uma fonte personificada de injustiça e tristeza no mundo (BALE, 2007) e podem levar a percepções equivocadas da realidade (LEWANDOWSKY *et al.*, 2017). Ambos os conceitos têm exemplos reais derivados de conteúdo que pode ser perfeitamente verdadeiro.

### 2.1.2 Mecanismos da Desinformação

Em suma, podemos posicionar o conceito de *fake news* facilmente dentro da literatura que envolve desinformação. Se esta desinformação é exibida em um formato jornalístico, ela é caracterizada como notícia falsa. Percepções equivocadas da realidade são consequência direta da exposição a notícias falsas. Rumores e teorias conspiratórias podem ser resultado tanto de conteúdo verídico como não-verídico. As notícias falsas podem (e são) utilizadas para propósitos propagandísticos políticos. A Figura 2 demonstra a relação e os mecanismos dos conceitos discutidos até aqui.

Figura 2 – Relações entre os conceitos associados à desinformação

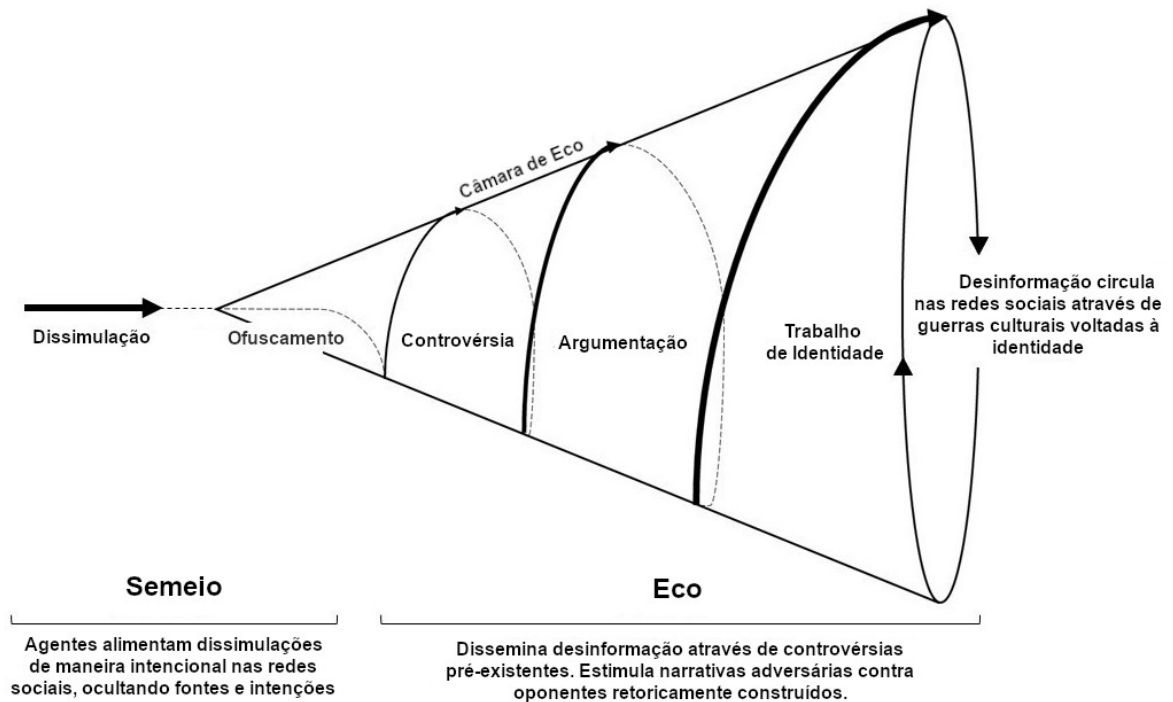


Fonte: adaptado de Egelhofer e Lecheler (2019, p. 13)

Ruiz C. (2022) define um mecanismo de duas fases que conceitualizam a disseminação de desinformação nas redes sociais: a primeira fase é o **semeio** ou **semeadura**, onde atores inserem de maneira estratégica informações enganosas, mascarando ou ofuscando declarações como se estas fossem legítimas. Semear neste conceito significa plantar ou inserir narrativas contraditórias ocultando-se a fonte, descontextualizar informações, embutir opiniões pessoais e camuflá-las como conversas legítimas em fontes de informação ditas confiáveis. A segunda fase, o dito **eco**, consiste em como a desinformação circula e se propaga nas câmaras de eco através de contradições contra oponentes em guerras culturais e outras controvérsias identitárias. Os

agentes do processo utilizam-se de falsidades, verdades seletivas, crenças, juízos de valor e todo o leque de controvérsias disponíveis para representar e explorar retoricamente sua identidade em oposição a oponentes conhecidos.

Figura 3 – Mecanismo de *eco* da desinformação



Fonte: adaptado de Ruiz C. (2022, p. 13)

## 2.2 Modelagem Semântico-Distribucional

A linguagem natural é como nós, humanos, trocamos ideias e opiniões. Existem dois meios principais para a linguagem natural: a oral e a escrita. O ato de ouvir e ler são facilmente executados por um ser humano saudável, mas são difíceis para um algoritmo computacional (TREVISIO *et al.*, 2022). Neste sentido, o advento das técnicas da área de Processamento de Linguagem Natural fazem bastante sentido: elas permitem que computadores possam processar linguagem humana e entender seu significado e contexto, bem como o sentimento associado e a intenção por trás dela, utilizando de suas descobertas para criar algo novo. É uma área que combina linguística computacional com modelos estatísticos de aprendizado de máquina e aprendizado profundo. Entretanto, um desafio surge: como fazer palavras serem interpretadas por máquinas computadoradas?

**Vetorização** é um jargão para uma abordagem clássica de conversão de dados de entrada de seu formato bruto (ou seja, textual) em vetores de números reais, que é o formato

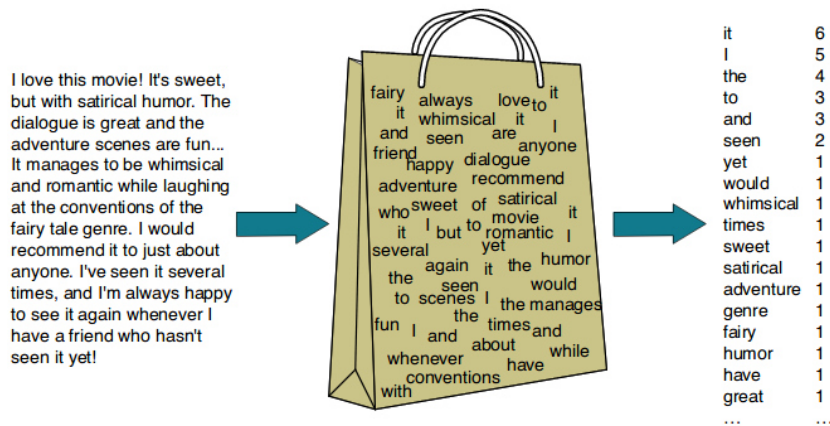
compatível com modelos de aprendizado de máquina. Essa abordagem existe desde os primórdios da invenção dos computadores, tem funcionado bem em diversos domínios e agora é uma das partes mais importantes da área de Processamento de Linguagem Natural. No campo de aprendizado de máquina, a vetorização é um passo na etapa de extração de características. A ideia é obter características distintas do texto para o modelo que será treinado, por meio da conversão de texto para vetores numéricos (JHA, 2022).

Existem diversas maneiras de se executar vetorização, como veremos em breve, desde simples recursos de ocorrência de termos binários até representações avançadas de recursos sensíveis ao contexto. Dependendo do caso de uso e do modelo, qualquer um deles pode realizar a tarefa necessária.

### 2.2.1 *Bag of Words*

Esta é a mais simples das técnicas de vetorização. Nela, um texto (como uma frase ou um documento) é representado como o “*saco*” de suas palavras, desconsiderando a gramática e até mesmo a ordem, porém mantendo a *multiplicidade* (HARRIS, 1954).

Figura 4 – Representação intuitiva do formato *bag of words*



Fonte: adaptado de Kumar (2021)

Essa abordagem geralmente é composta de três operações:

1. **Tokenização:** O texto de entrada é separado em *tokens*. Uma sentença é representada como a lista de suas palavras constituintes, e esse processo é feito para todas as sentenças de entrada;
2. **Criação de Vocabulário:** A partir de todas as palavras tokenizadas obtidas, apenas palavras únicas são selecionadas para criar o vocabulário e, em seguida,

classificadas por ordem alfabética;

3. **Criação de Vetor:** Finalmente, uma matriz esparsa é criada para a entrada, a partir da frequência do vocabulário de palavras. Nessa matriz, cada linha é um vetor de sentença cujo comprimento (as colunas) é igual ao tamanho do vocabulário.

### 2.2.2 TF-IDF

TF-IDF ou Frequência do Termo - Frequência Inversa dos Documentos é uma métrica estatística que tem a intenção de refletir o quão importante uma palavra é para um documento inteiro. Apesar de ser outro método baseado em frequência, não é tão rudimentar quanto o *bag of words*.

Na abordagem do saco de palavras, a preocupação com a frequência de palavras do vocabulário pauta totalmente a representação numérica do documento. Isto quer dizer que artigos, preposições e conjunções, os quais não contribuem tanto assim para o significado, ganham tanta importância quanto adjetivos, por exemplo. O TF-IDF ajuda a superar essa limitação. Palavras que são repetidas demais não ofuscam palavras menos frequentes mas ainda assim importantes. Possui duas partes:

1. **Frequência de Termos:** Pode ser compreendida como um escore de frequência normalizado. É calculado a partir a seguinte fórmula:

$$TF(t, d) = \frac{\text{número de vezes que } t \text{ aparece em } d}{\text{número total de termos em } d} \quad (2.1)$$

2. **Frequência Inversa dos Documentos:** é necessário definir antes sua componente, DF. É dada pela seguinte fórmula:

$$IDF(t) = \log \frac{N}{1 + df} \quad (2.2)$$

onde  $df$  é a frequência dos documentos, que afirma a proporção de documentos que possuem uma certa palavra. O termo  $df$  é invertido porque quanto mais comum uma palavra é em todos os documentos, menos é sua importância para o documento atual. O logaritmo é usado para atenuar o efeito de IDF no cálculo final.

O escore final TF-IDF é então definido como:

$$TFIDF(t, d) = TF(t, d) \cdot IDF(t) \quad (2.3)$$

Essa é a maneira como o cálculo TF-IDF consegue incorporar o significado de uma palavra. Quanto maior o valor do escore, mais importante é a palavra.

### 2.3 Sistemas de Recomendação

Um **sistema recomendador** ou **sistema de recomendação** (podemos aqui entender sistema como “plataforma” ou “motor”) é um conjunto de técnicas e ferramentas de software que provém sugestões para itens que são mais pertinentes a um usuário em particular. Em termos gerais, as sugestões referem-se a vários processos de tomada de decisão, como exemplo, qual item comprar, qual música ouvir ou mesmo qual notícia ler. Esses sistemas são particularmente úteis quando um único indivíduo precisa fazer a escolha de um item num conjunto extraordinariamente grande de escolhas que um serviço pode ofertar (RICCI *et al.*, 2022).

Os sistemas de recomendação são ferramentas que possibilitam a interação com espaços de informação grandes e complexos. Eles provém uma visão personalizada desses espaços, dando prioridade a itens que são de provável interesse ao usuário. A área de pesquisa tem crescido enormemente no que concerne à variedade de problemas e técnicas empregadas, bem como as suas aplicações práticas (BURKE *et al.*, 2011). Exemplos dessas aplicações mais reconhecíveis são os geradores de *playlists* de serviços de vídeo e música, recomendadores de produtos para lojas *online*, ou recomendadores de conteúdo para plataformas de rede social, bem como recomendadores de conteúdo na *web* aberta (BARAN *et al.*, 2018). Ainda segundo Ricci *et al.* (2022), os dados de um sistema recomendador podem se relacionar com três tipos de entidades: *itens*, *usuários* e *interações* entre os usuários e os itens.

- **Itens:** são os objetos a serem recomendados. A disponibilidade dos dados dos itens dependerá da complexidade de aquisição. Por exemplo, em um sistema recomendador multimídia, itens como imagens requerem algoritmos especiais de análise de imagem para que seja possível a extração de características a partir do conteúdo bruto. Conteúdos baseados em texto, como notícias, análises ou itens que possuem descrições longas requerem algoritmos especiais para a análise do texto e a extração de características relevantes do mesmo.

- **Usuários:** em um sistema de recomendação, eles podem ter diversos objetivos e características. Visando a personalização das recomendações e da interação humano-computador, os sistemas recomendativos exploram um certo conjunto de informações a respeito dos usuários. Esta informação pode ser estruturada de diversas maneiras. De acordo com (BILLSUS; PAZZANI, 1997) e (FISCHER, 2001), os dados do usuário definem a modelagem do usuário. O modelo traça o perfil do usuário, isto é, codifica suas necessidades e preferências. Diversas abordagens de modelagem de usuário tem sido utilizadas e, em um certo sentido, um sistema recomendador pode ser encarado como uma ferramenta que gera recomendações ao construir e explorar perfis de usuário (BERKOVSKY *et al.*, 2008) (BERKOVSKY *et al.*, 2009).
- **Interações:** elas contemplam as associações entre usuários e itens. Elas são registradas em dados em formato de histórico que armazena informações importantes geradas durante processos que estabeleçam relações interativas entre as duas entidades anteriores. Esta informação é útil para o algoritmo de geração de recomendação que o sistema utiliza. Por exemplo, um registro poderá conter a referência ao item comprado pelo usuário, os eventos que levaram ao evento de compra (passos de navegação, cliques, inclusão no carro de compras etc), a avaliação que o usuário atribuiu ao item, qualquer outra informação que o sistema coleta sobre o tempo ou a localização da interação, entre outros atributos. A parte mais importante da interação é o *feedback* dos usuários, que pode ser explicitamente fornecido por eles ou implicitamente inferido a partir dos seus comportamentos (RICCI *et al.*, 2022).

### 2.3.1 Formalização do Problema

A definição formal do problema de recomendação pode variar conforme a perspectiva dos autores na literatura, porém, será aqui adotado a abordagem da pesquisa de levantamento de (ADOMAVICIUS; TUZHILIN, 2005): em um certo sistema, seja  $C$  o conjunto de todos os usuários e seja  $S$  o conjunto de todos os itens que podem ser recomendados. Seja  $u$  uma função utilidade que mede o quão *útil* um item  $s$  é para um usuário  $c$ , isto é,  $u: C \times S \rightarrow R$ , onde  $R$  é um conjunto totalmente ordenado (isto é, inteiros não-negativos ou números reais dentro de um certo intervalo). Então, para cada usuário  $c \in C$ , queremos escolher itens  $s' \in S$  que maximizem a utilidade para o usuário. Em termos matemáticos,

$$\forall c \in C, s'_c = \arg \max_{s \in S} u(c, s) \quad (2.4)$$

Ainda segundo (ADOMAVICIUS; TUZHILIN, 2005), a utilidade de um item é geralmente representada por uma avaliação que indica o quanto um usuário em particular "gostou" de um item específico. Entretanto, como indicado anteriormente, no geral, a utilidade pode ser uma função arbitrária. Dependendo da aplicação, a utilidade  $u$  pode ou ser especificada pelo usuário, como é geralmente feito em avaliações definidas por ele, ou calculada pela aplicação, como é feito para o caso de uma função de utilidade baseada em *lucro*.

Cada elemento do conjunto  $C$  de usuários pode ser definido por um *perfil* que inclui diversas características de usuário, tais como idade, gênero, salário, estado civil, entre outras. De maneira similar, cada elemento do conjunto de itens  $S$  é definido por um grupo de características. Por exemplo, em uma tarefa de recomendação de filmes, cada filme poderá ser representado não apenas por seu identificador, mas como também pelo seu título, gênero, diretor, ano de lançamento, etc.

O problema central referente aos sistemas de recomendação reside no fato de que, geralmente, a utilidade  $u$  não é definida sob todo o espaço  $C \times S$ , mas em apenas um *subconjunto* dele. Em outras palavras, isto significa que  $u$  precisa ser *extrapolada* para todo o espaço  $C \times S$ . As problemáticas geradas por essa eventual incompletude do conjunto  $C \times S$  serão exploradas mais adiante.

### 2.3.2 Modelagens e Técnicas de Recomendação

De acordo com (RICCI *et al.*, 2022), para que a função principal de um sistema recomendador seja implementada, isto é, a identificação de itens úteis ao usuário, é preciso que o sistema possa *prever* o que deve ser recomendado. Tendo em vista esta tarefa, o sistema deverá ser capaz de prever a utilidade de alguns dos itens, ou pelo menos comparar a utilidade de alguns deles, e só então decidir quais itens recomendar baseado nesta comparação.

Para apresentar uma visão geral inicial dos diferentes tipos de sistemas de recomendação, será aqui mencionada uma visão mais clássica, desenvolvida por (BURKE, 2007), que distingue as abordagens em seis classes diferentes de recomendação:

1. Filtragem colaborativa;
2. Filtragem baseada em conteúdo;

3. Filtragem baseada em comunidade;
4. Filtragem demográfica;
5. SR baseados em conhecimento;
6. SR Híbridos.

Segundo Aggarwal *et al.* (2016), os modelos mais básicos de sistemas de recomendação trabalham com dois tipos de dados, os quais são:

1. as interações usuário-item, tais como avaliações ou comportamento consumidor;
2. a informação dos atributos dos usuários e dos itens tais como perfis textuais ou palavras-chave relevantes.

Métodos que utilizam a primeira abordagem são usualmente referidos como métodos de *filtragem colaborativa*, enquanto que os que utilizam a segunda são normalmente chamados de métodos *baseados em conteúdo*. Nos sistemas de recomendação *baseados em conhecimento*, as recomendações são fundamentadas em requisitos de usuário explicitamente especificados.

Em vez de dados históricos ou dados de compra, bases externas de conhecimento e restrições são utilizadas para construir a recomendação. Alguns desses sistemas recomendadores combinam esses diferentes aspectos para criar *sistemas híbridos*. Esses últimos são capazes de combinar os pontos fortes de diversos tipos de sistemas de recomendação para criar técnicas que podem desempenhar de maneira mais robusta em uma grande variedade de configurações. Esta seção propõe fazer um destrinchamento teórico de três destes métodos, uma vez que estes são mais pertinentes e expressivos no que concerne ao trabalho desta pesquisa.

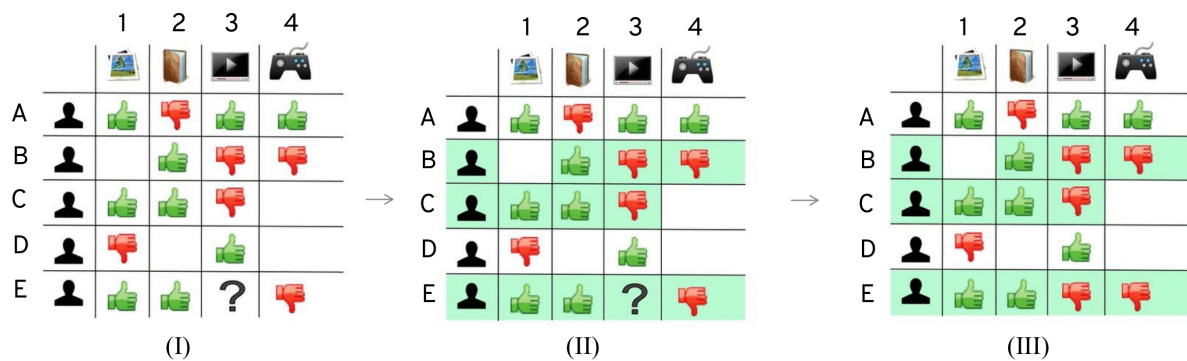
### 2.3.2.1 Filtragem Colaborativa

Sistemas de Recomendação que empregam a abordagem de *Filtragem Colaborativa* utilizam o poder cooperativo de avaliações fornecidas por múltiplos usuários para realizar recomendações (AGGARWAL *et al.*, 2016). A ideia básica da filtragem colaborativa consiste no fato de que, se uma pessoa A tem a mesma opinião que uma pessoa B sobre um certo assunto, é mais *provável* que A tenha uma opinião bem mais parecida com a de B em um assunto diferente do que uma outra pessoa escolhida aleatoriamente. Por exemplo, um sistema que emprega essa técnica, utilizado no contexto de preferências de programação para TV, poderia fazer previsões sobre qual programa de televisão um usuário deve gostar, dada uma lista parcial dos gostos (ou desgostos) desse usuário.

A Figura 5 demonstra um passo-a-passo intuitivo desenvolvido em torno desta ideia

base: em I queremos prever se o usuário E irá ou não gostar do item 3. Em II, fazendo-se uma avaliação objetiva dos gostos dos outros usuários, percebemos maior semelhança de avaliações em B e C (em relação a E, A possui uma avaliação igual, B e C possuem duas avaliações iguais, D tem nenhuma avaliação igual). Em III, ao selecionarmos os usuários mais semelhantes B e C, podemos prever que se ambos “não gostaram” de 3, podemos fazer a predição “não gosto” do usuário E para o item 3.

Figura 5 – Etapas da filtragem colaborativa



Fonte: adaptado de Kulkarni (2017, p.7-8)

Uma das grandes vantagens que a filtragem colaborativa possui é que ela não requer informações subjetivas nem um entendimento mais profundo sobre usuários ou itens, assim podem ser aplicados em muitas situações. Além disso, quanto mais usuários interagem com itens, mais novas recomendações se tornam mais precisas: para um conjunto fixo de usuários e itens, novas interações registradas ao longo do tempo trazem novas informações e tornam o sistema mais e mais efetivo.

De acordo com (AGGARWAL *et al.*, 2016), sistemas de recomendação que aplicam a abordagem de filtragem colaborativa possuem dois tipos de métodos que são comumente empregados: os métodos baseados em *memória* e os métodos baseados em *modelo*.

- **Métodos baseados em memória:** também chamados de *algoritmos de filtragem colaborativa baseados em vizinhanças*, estão entre os primeiros algoritmos de filtragem colaborativa. Neles, as avaliações provindas das combinações usuário-item são previstas de acordo com as suas vizinhanças. Essas vizinhanças podem ser lidadas de duas maneiras: uma perspectiva voltada para *usuário* e outra para *item*.

1. *Filtragem colaborativa baseada em usuário:* as avaliações fornecidas por usuários com interesses semelhantes a um usuário-alvo A são usadas para fazer as recomendações para A e recomendar avaliações para as ainda

não observadas de A, calculando, desta forma, médias ponderadas das avaliações deste grupo de pares. O exemplo anterior da Figura 5 apresenta uma ideia rudimentar, porém bastante similar a esta perspectiva.

2. *Filtragem colaborativa baseada em item*: nesta perspectiva, a fim de se fazer previsões de avaliações para um item-alvo I por um usuário A, o primeiro passo é determinar um conjunto  $S$  de itens que são os mais similares ao item-alvo I. As avaliações no conjunto de itens  $S$ , que são especificadas pelo usuário A, são utilizadas para prever se o usuário A irá gostar do item I. Portanto, as avaliações de Alice em filmes de ficção científica como *Alien vs. Predator* podem ser utilizadas para prever sua avaliação em *Terminator*. Para a primeira perspectiva, funções de similaridade são computadas entre as *linhas* das matrizes de avaliação para descobrir *usuários* similares. Para a segunda, essas funções são computadas entre as *colunas* para a descoberta de *itens* similares.

– **Métodos baseados em modelo**: aplicam técnicas de aprendizado de máquina e mineração de dados no contexto de modelos preditivos. Nos casos em que o modelo é parametrizado, os parâmetros desse modelo são aprendidos no contexto de um *framework* de otimização. Alguns exemplos de tais métodos baseados em modelos incluem árvores de decisão, modelos baseados em regras, métodos Bayesianos e modelos de fatores latentes.

Ainda segundo (AGGARWAL *et al.*, 2016), sistemas de recomendação baseados em modelo possuem geralmente algumas vantagens sobre os baseados em vizinhança/memória: em primeiro lugar, eles possuem *eficiência espacial*, pois o tamanho do modelo aprendido é muito menor que uma matriz de avaliações. Com isso, os requisitos são geralmente bem baixos. Por outro lado, em um método baseado em memória convencional, se  $n$  representa o número de usuários (ou de itens), ele pode possuir complexidade espacial  $O(n^2)$ .

Em segundo lugar, eles apresentam maior *velocidade de treinamento e de predição*: um dos problemas de métodos de vizinhança decorre do fato que o estágio de pré-processamento, conforme mencionado anteriormente, é quadrático em termos de usuários ou de itens. Sistemas baseados em modelos são geralmente bem mais rápidos na fase de pré-processamento da construção do modelo treinado. Na maioria das vezes, uma forma simplificada e compacta do eventual modelo pode ser utilizada para se fazer previsões de maneira bastante eficiente.

Por último, eles buscam prevenir *overfitting*, isto é, o sobreajuste do modelo aos

dados de treinamento. *Overfitting* é um grande obstáculo em muitos modelos de aprendizagem de máquina, nos quais a predição é gravemente afetada por artefatos aleatórios nos dados. Problemas de classificação e regressão frequentemente lidam com este inconveniente. Algumas soluções para esta problemática envolvem a abordagem de sumarização de métodos baseados em modelo e a de *regularização*, que tornam os modelos mais robustos e resistentes a enviesamento e sobreajuste.

É importante ressaltar e fácil de notar que uma das maiores problemáticas da Filtragem Colaborativa decorre do fato de as matrizes de avaliações geradas serem *esparsas*. Na verdade, na maioria das vezes, em grande parte das avaliações de um sistema recomendador real, elas não são especificadas (AGGARWAL *et al.*, 2016). Isto ocasiona o chamado *começo frio* (ou *cold start problem* como declara a literatura), que descreve a dificuldade da filtragem colaborativa em se fazer recomendações quando os usuários ou os itens são novos. Como o método dessa abordagem de recomendação se baseia nos dados históricos das interações usuário-item, num cenário de estágio inicial do sistema, novos usuários simplesmente não possuem dados históricos o suficiente para que ele possa funcionar adequadamente (SKOVHØJ, 2022). Outro desafio de sistemas que implementam este paradigma foi mencionado anteriormente: a *escalabilidade*. À medida em que o número de itens e de usuários cresce, algoritmos tradicionais de filtragem colaborativa sofrem sérios problemas de performance simplesmente devido ao grande aumento na quantidade de dados. Ainda segundo (SKOVHØJ, 2022), esses sistemas são *incapazes de identificar sinônimos*, ou seja, itens semelhantes rotulados ou nomeados de maneira diferente. A filtragem colaborativa é incapaz de descobrir a associação latente entre sinônimos, pois ela analisará os itens diferentemente. Por exemplo, um sistema dessa natureza seria incapaz de determinar que "bolsa" e "mochila" na verdade poderiam significar um mesmo item.

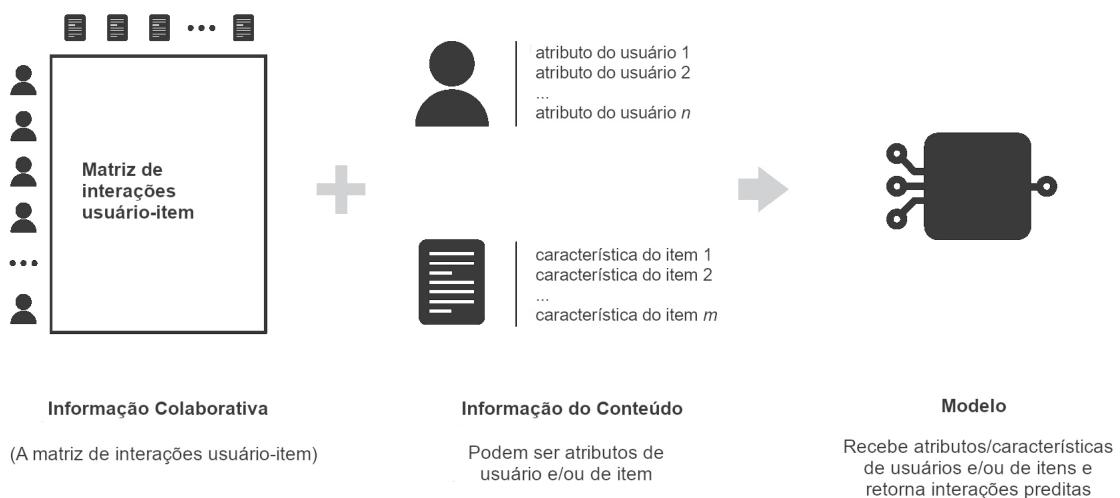
### 2.3.2.2 Filtragem Baseada em Conteúdo

No contexto de sistemas de recomendação, *filtragem baseada em conteúdo* é um tipo de abordagem que é fundamentada em uma descrição do item e um perfil das preferências do usuário (AGGARWAL *et al.*, 2016) (BRUSILOVSKI *et al.*, 2007). O termo *conteúdo* se refere a essas descrições. Esses métodos são mais adequados em situações onde os itens e seus atributos são bem conhecidos (nome, local, descrição etc) mas o usuário não é. Os métodos orientados a conteúdo lidam com a tarefa de recomendação como um problema de classificação específico do usuário, e tentam aprender as preferências dele baseados nos atributos dos itens.

O sistema aprende a recomendar itens que são similares aos que o usuário aprovou no passado. A similaridade dos itens é então calculada com base nos atributos associados aos itens em comparação. Por exemplo, se um usuário avaliou positivamente um filme que pertence ao gênero de comédia, então o sistema poderá aprender a recomendar outros filmes deste mesmo gênero (LOPS *et al.*, 2011). Na maioria dos casos deste tipo de abordagem, palavras-chave são utilizadas para a descrição dos itens, e um **perfil de usuário** é criado para indicar o tipo de item que esse usuário gosta.

Um *perfil de usuário* é uma coleção de definições e informações associadas ao usuário, caracterizando-o. Para criar este perfil, o sistema se concentra em dois tipos de informação: um modelo das preferências do usuário e um histórico das interações do usuário com o sistema recomendador. Em um contexto de um sistema que recomenda filmes, por exemplo, essa modelagem do perfil poderá ser entendida como o fato de que mulheres mais jovens tem a tendência a avaliar positivamente certos filmes, enquanto que homens jovens tem esta mesma tendência com filmes diferentes, e assim em diante. A Figura 6 apresenta uma visão de alto nível do paradigma de sistema recomendador orientado a conteúdo.

Figura 6 – Visão geral do paradigma orientado a conteúdo



Fonte: adaptado de Rocca (2019)

De acordo com Aggarwal *et al.* (2016), sistemas de recomendação com métodos baseados em conteúdo possuem algumas vantagens na recomendação de um **novo item** quando não há avaliações suficientes a respeito dele. Isto se deve ao fato de que outros itens com características similares podem já ter sido avaliados por usuários ativos. Desta maneira, o modelo supervisionado será capaz de se aproveitar destas avaliações de maneira conjunta com os atributos

do novo item para fazer recomendações mesmo que não exista histórico de avaliações desse item. Segundo Ricci *et al.* (2022), esses sistemas também tem ***independência de usuário***: sistemas de recomendação baseados em conteúdo exploram exclusivamente avaliações fornecidas por um usuário ativo para que seu próprio perfil seja construído. Em contrapartida, métodos de filtragem colaborativa precisam de avaliações de outros usuários para encontrar os vizinhos mais próximos do usuário ativo ou até mesmo precisam construir modelos sofisticados de aprendizado de máquina. Em outras palavras, sistemas de recomendação orientados por conteúdo são menos suscetíveis a problemas de *esparsidade dos dados* e podem ser mais efetivos quando há pouca disponibilidade de dados em mãos. Ricci *et al.* (2022) também argumenta que sistemas desse paradigma são ***transparentes***, pois explicações sobre como o sistema funciona podem ser obtidas listando explicitamente as características e atributos do conteúdo ou descrições que motivaram a ocorrência de um item na lista de recomendações. Essas características são indicadores de consulta para que se possa decidir a respeito da confiabilidade da recomendação. Por outro lado, sistemas colaborativos são como caixas-pretas, pois o único detalhamento que se pode obter a respeito de uma recomendação de item é que usuários desconhecidos com gostos similares gostaram daquele item.

Ainda segundo Aggarwal *et al.* (2016), este paradigma também possui algumas desvantagens. A primeira delas é que, em muitos casos, métodos baseados em conteúdo fornecem ***recomendações óbvias*** por causa do uso de palavras-chave ou conteúdo em si. Por exemplo, se um usuário nunca consumiu um item com um conjunto particular de palavras-chave, esse item não terá chance alguma de ser recomendado. Isto ocorre porque o modelo construído é específico para o usuário em questão e ele não é capaz de aproveitar conhecimento adquirido de usuários similares dentro do conjunto de dados comunitário. A segunda desvantagem é que, apesar dos métodos deste paradigma serem efetivos em prover recomendações para *novos itens*, eles não são efetivos em prover recomendações para ***novos usuários***. Isto acontece porque o modelo em treinamento para o usuário-alvo precisa ter acesso ao seu histórico de avaliações. Na verdade, é importante se ter um número grande de avaliações disponíveis para o usuário-alvo para que seja possível fazer predições robustas sem enviesamento. Por fim, conforme é dito por (RICCI *et al.*, 2022), métodos baseados em conteúdo possuem uma ***análise de conteúdo limitada***, isto é, têm naturalmente um limite no número e no tipo de características associadas, seja automaticamente ou manualmente, com os objetos que eles recomendam. Nenhum sistema de recomendação baseado em conteúdo pode fornecer sugestões adequadas se não houver recursos descritivos dos

itens disponíveis.

### 2.3.2.3 Modelos Híbridos

Grande parte dos sistemas de recomendação em uso atualmente usam uma abordagem híbrida, isto é, combinam filtragem colaborativa, filtragem com base em conteúdo e outras abordagens. Na verdade, não há razão inerente pela qual diferentes técnicas do mesmo tipo não possam ser hibridizadas; de fato, conforme argumenta (RICCI *et al.*, 2022), um sistema híbrido combinando técnicas A e B tenta usar as vantagens de A para corrigir as desvantagens de B. Por exemplo, métodos de filtragem colaborativa sofrem com problemas de introdução de novos itens, isto é, eles são incapazes de recomendar itens que ainda não possuem avaliações. Isto não prejudica o fluxo dos métodos orientados a conteúdo uma vez que a predição para novos itens é baseada nas descrições (características) dos itens que normalmente estão facilmente disponíveis.

Existem diversas formas de implementação destas combinações, desta forma, seu apelo teórico e suas peculiaridades podem variar de acordo com a literatura e o contexto aplicado. Se levarmos em consideração os sistemas de filtragem colaborativa e os voltados para conteúdo, tendo em vista uma implementação de um modelo híbrido, (ADOMAVICIUS; TUZHILIN, 2005) definem as seguintes formas de execução:

1. Implementação de métodos colaborativos e baseados em conteúdo separados entre si, para depois combinar suas predições;
2. Incorporação de alguns aspectos de orientação a conteúdo em uma abordagem colaborativa;
3. Incorporação de algumas características colaborativas em uma abordagem baseada em conteúdo;
4. Construção de um modelo geral unificador que incorpora ambas as características.

Diversos estudos que comparam empiricamente o desempenho de modelos híbridos com métodos puramente colaborativos ou baseados em conteúdo demonstram que os métodos híbridos podem fornecer recomendações mais precisas do que as abordagens puras. (TIEMANN; PAUWS, 2007) propõem um método colaborativo baseado em memória e item e um método baseado em conteúdo de áudio que faz predições de avaliações independentes, as quais são agregadas com base em uma avaliação de vetor de similaridades. (LU; TSENG, 2009) fundem a saída de três recomendadores de vizinhos mais próximos: dois com abordagem baseada em conteúdo que computam a similaridade de partituras musicais e classes de emoções, e um com

abordagem colaborativa; a lista de recomendação final é então criada ao se repriorizar itens com base na ponderação personalizada das três componentes. (MCFEE *et al.*, 2012) otimizam uma métrica de similaridade baseada em conteúdo com a etapa de aprendizado que utiliza amostras de dados colaborativos. Por fim, o exemplo prático demonstrado por (GOMEZ-URIBE; HUNT, 2016) afirma que a Netflix<sup>1</sup>, popular serviço de *streaming* de conteúdo televisivo, faz recomendações comparando os hábitos de visualização e pesquisa de usuários semelhantes (isto é, filtragem colaborativa) e também oferece filmes que compartilham características com outros filmes que um usuário avaliou positivamente (filtragem baseada no conteúdo).

## 2.4 Funções de Similaridade

O conceito de similaridade é bastante subjetivo e possui uma grande dependência do contexto e da aplicação. De acordo com Kuneš (2012), medidas estatísticas de similaridade permitem que pesquisadores possam pensar matematicamente e computacionalmente sobre o quão parecidos ou diferentes seus objetos de estudo possam ser, e essas medidas de fato se tornam os blocos de construção de muitas outras técnicas de agrupamento e classificação.

No campo estatístico e outros afins, uma **métrica de similaridade** ou uma **função de similaridade** é uma função de variável real que quantifica a similaridade entre dois objetos. Embora não exista uma definição única de similaridade, geralmente essas medidas são de certa forma a inversa das métricas de distância: grandes valores representam objetos bastante semelhantes e valores nulos ou negativos representam objetos que são muito diferentes. Contudo, em termos mais genéricos, uma função de similaridade também pode satisfazer axiomas métricos. Em um contexto de mineração de dados, a medida de similaridade representa a distância de cada uma das dimensões que descrevem características de um objeto ou entidade. No campo da análise textual, a similaridade entre dois textos pode ser avaliada na sua forma mais básica ao se representar cada texto como uma série de contagem de palavras e ao se calcular a distância usando essa contagem como característica ou representação do texto. No contexto deste trabalho, que é o de sistemas de recomendação, a similaridade poderá ser utilizada como esta propriedade que atestará o quão uma entidade, leia-se *usuário* ou *item*, é mais parecida (ou não) com outra.

Ao considerarmos essa faceta ambígua dessa definição do conceito de similar, não será incomum verificarmos exemplos de que essas métricas de similaridade compreendem conceitos tecnicamente distintos, mas que possuem uma intersecção na representação abstrata

<sup>1</sup> <https://www.netflix.com>

da entidade em termos de distância: a similaridade entre dois *itens* de conteúdo textual de um sistema de recomendação poderá também ser vista como uma simples similaridade entre dois conteúdos textuais propriamente ditos, ou seja, encarada como a comparação matemática direta entre suas contagens de palavras, por exemplo. Nesta seção será delineada a base teórica de um conjunto selecionado de métricas de similaridade. Para cada métrica, será apresentada uma breve descrição e sua fórmula matemática correspondente.

#### 2.4.1 Distância Euclidiana

A Distância Euclideana, denotada por  $d$ , é uma medida amplamente usada como uma métrica tradicional de distância em trabalhos com geometria (O'NEILL, 2006). Pode-se entendê-la como uma distância normal entre dois pontos. É uma das métricas mais utilizadas em análise de agrupamentos: o próprio algoritmo K-Médias baseia-se no conceito de distância euclidiana para gerar seus grupos (NA *et al.*, 2010). Ela pode ser entendida como a Equação 2.5:

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^n |x_i - y_i|^2} \quad (2.5)$$

Para que a distância euclideana se torne uma métrica de similaridade, ela precisa ser normalizada. Se  $\mathbf{x}$  e  $\mathbf{y}$  são a representação vetorial de *usuários* ou de *itens* e  $d(\mathbf{x}, \mathbf{y})$  representa a distância euclideana, então a Equação 2.6 define a Similaridade Euclideana entre as duas entidades:

$$ES(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + d(\mathbf{x}, \mathbf{y})} \quad (2.6)$$

#### 2.4.2 Similaridade do Cosseno

Matematicamente, a similaridade do cosseno é a medida de similaridade entre dois vetores não-nulos na forma de um produto interno em um espaço vetorial que mede o cosseno do ângulo entre eles (BREESE *et al.*, 2013). Ou seja, o valor do cosseno dos dois vetores que representam dois usuários (ou itens) indica o valor da similaridade entre eles. Se  $\mathbf{x}$  e  $\mathbf{y}$  são vetores, a similaridade do cosseno entre eles é definida pela Equação 2.7:

$$\text{cosseno}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}^T}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (2.7)$$

O cosseno de 0 é 1, e será menor que 1 para qualquer ângulo no intervalo  $(0, 2\pi]$  radianos. Trata-se, portanto, de um julgamento de *orientação* e não de *magnitude*: dois vetores com a mesma orientação possuem a similaridade do cosseno igual a 1, dois vetores orientados em  $90^\circ$  relativamente entre si possuem similaridade igual a zero, e dois vetores diametralmente opostos possuem similaridade de -1, independentemente das suas magnitudes. Geralmente, a similaridade do cosseno é particularmente utilizada no espaço dos números reais não-negativos, onde o valor é nitidamente limitado por  $[0, 1]$ .

Uma vantagem da similaridade do cosseno é sua *baixa complexidade*, especialmente para *matrizes esparsas*: apenas as coordenadas não-nulas precisam, de fato, serem consideradas. Em um contexto de filtragem colaborativa, um problema com essa métrica é que ela não leva em consideração as diferenças na média e na variância das prováveis avaliações feitas por dois usuários.

### 2.4.3 Correlação de Pearson

Esta métrica foi inicialmente proposta por Karl Pearson (PEARSON, 1895) para se mensurar relações lineares e se tornou amplamente utilizada no campo estatístico. O coeficiente de correlação de Pearson calcula a correlação linear entre *duas variáveis aleatórias conjuntamente* (MAO, 2021). Amostramos  $n$  amostras de uma distribuição conjunta bivariada  $(X, Y)$ . Dadas  $n$  amostras que consistem de duas variáveis,  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , e se  $cov$  é a covariância,  $\sigma_X$  e  $\sigma_Y$  o desvio padrão de  $X$  e  $Y$  respectivamente, o coeficiente de Correlação de Pearson é definido como

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \quad (2.8)$$

$$= \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (2.9)$$

$$= \frac{\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]}{\sqrt{\mathbb{E}[X^2] - (\mathbb{E}[X])^2} \sqrt{\mathbb{E}[Y^2] - (\mathbb{E}[Y])^2}} \quad (2.10)$$

Quando o coeficiente de correlação de Pearson é aplicado sobre uma amostra,

podemos encontrar a representação que é normalmente utilizada em sistemas de recomendação: esta forma é normalmente representada por  $r(x,y)$  e pode ser referida como *coeficiente de correlação amostral* ou *correlação de Pearson de amostra*. Podemos obter a fórmula para  $r(x,y)$  ao substituir as covariâncias e variâncias esperadas com base em uma amostra na Equação 2.10, onde  $x$  e  $y$  representam as entidades (itens ou usuários) que serão comparados:

$$r(x,y) = \frac{(\sum_{i=1}^n x_i y_i) - (n \bar{x} \bar{y})}{\sqrt{\sum_{i=1}^n x_i^2 - n \bar{x}^2} \sqrt{\sum_{i=1}^n y_i^2 - n \bar{y}^2}} \quad (2.11)$$

Esta função retorna um valor no intervalo  $[-1, 1]$ , onde 1 representa forte correlação positiva, -1 uma forte correlação negativa, e 0 absolutamente nenhuma correlação.

#### 2.4.4 Índice de Jaccard

O índice de Jaccard (JACCARD, 1912), denotado por  $J$ , também conhecido como *coeficiente de similaridade de Jaccard*, é uma métrica estatística usada para medir a similaridade e a diversidade de conjuntos. O coeficiente de Jaccard entre dois conjuntos finitos é definido como a cardinalidade da interseção dividida pela cardinalidade da união. Isto significa dizer que ele mede a proporção do número de elementos compartilhados entre os dois conjuntos com o número total de elementos em ambos os conjuntos. O índice  $J$  possui o valor no intervalo  $[0, 1]$ , quanto mais próximo de um, mais similares os dois vetores são, e a lógica inversa vale para zero.

A Equação 2.12 computa o índice de Jaccard para dois vetores  $\mathbf{x}$  e  $\mathbf{y}$ , considerando que  $\mathbf{x}$  e  $\mathbf{y}$  podem ser usuários (um conjunto de avaliações atribuídas pelo mesmo usuário) ou itens (um conjunto de avaliações atribuídas ao mesmo item). Como se tratam de dois conjuntos, podemos definir o índice da seguinte forma:

$$J(\mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x} \cap \mathbf{y}|}{|\mathbf{x} \cup \mathbf{y}|} \quad (2.12)$$

A principal desvantagem (PATRA *et al.*, 2015) deste método é que ele não considera as avaliações absolutas e é extremamente sensível ao tamanho pequeno dos dados com observações ausentes.

## 2.5 Métricas de Avaliação de Desempenho

O processo de avaliação de performance de um sistema recomendador é crucial para obter-se um entendimento sobre a efetividade geral dos algoritmos desenvolvidos para o propósito de recomendação. Esse processo possui diversos aspectos e perspectivas, portanto um único critério não é capaz de contemplar muitos dos objetivos do desenvolvedor do sistema. Uma construção incorreta de uma avaliação experimental poderá levar à uma subestimação grosseira ou mesmo uma superestimação da verdadeira precisão de um determinado algoritmo ou modelo.

Segundo (AGGARWAL *et al.*, 2016), os métodos de avaliação de desempenho em sistemas recomendadores são geralmente divididos em duas categorias principais: **métodos online** e **métodos offline**. Métodos *online* avaliam suas recomendações obrigatoriamente com a participação dos usuários. Por exemplo, na avaliação de um sistema recomendador de notícias, seria possível aferir a taxa de interação de usuários com o número de cliques realizados em listas de artigos recomendados. Esses métodos são conhecidos como *testagem A/B*, e eles medem o desempenho direto do sistema recomendador para o usuário. Entretanto, métodos *online* possuem uma forte limitação de requerer participação direta do usuário, por isso não são adequados para o propósito de pesquisa nem avaliação comparativa. Para este intuito, os métodos *offline* fazem mais sentido, sendo quase sempre o ponto inicial de desenvolvimento de um processo avaliativo de um sistema de recomendação. Desta forma, a avaliação *online* é quase sempre um processo *posterior* à avaliação *offline*. Esta última será alvo deste tópico, tendo em vista as pretenções do trabalho desta pesquisa.

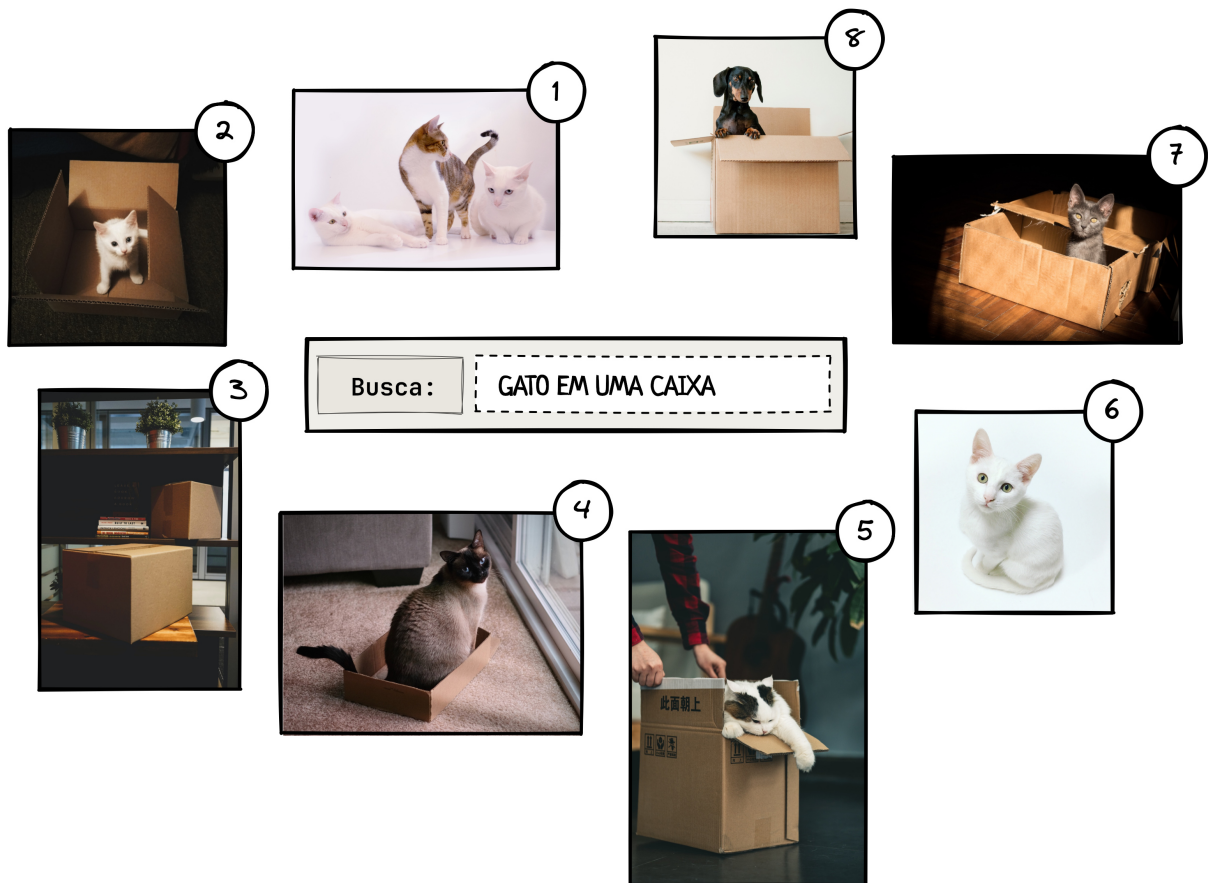
Quando trabalhamos com avaliações *offline*, métricas de precisão podem fornecer uma imagem incompleta da taxa verdadeira de aceitação de um sistema recomendador. Desta forma, é importante desenvolver o processo de avaliação cautelosamente para que as métricas de aferição reflitam de fato a efetividade de um sistema recomendador na perspectiva de um usuário. Geralmente, os métodos de avaliação *offline* são subdivididos em duas categorias: os métodos **sensíveis à ordem** e os **não-sensíveis à ordem**, ou seja, se a ordenação da recomendação dos itens é importante e deverá ter (ou não) impacto no resultado da avaliação. Em suma, para o primeiro método, a ordem dos itens **é importante no cálculo da relevância dos itens, enquanto que no segundo ela não importa.**

No processo de avaliação de um sistema de recomendação, as comparações são dadas entre itens preditos e itens "reais". Um item real neste contexto significa ser um item **relevante** de fato. É desejado que as condições preditas sejam quanto mais próximas possíveis

das condições reais. Para o propósito deste trabalho, tem-se o intuito de avaliar o ranqueamento de mensagens no que concerne à **utilidade** delas. Desta forma, serão demonstradas, de acordo com esse propósito de utilidade, métricas sensíveis (e insensíveis) à ordem que contemplem esse propósito.

Para demonstrar as métricas dispostas a seguir, consideremos doravante o exemplo empregado por Carnevali (2022): imagine um motor de busca hipotético com um conjunto de dados *bastante* pequeno: queremos recuperar a informação fornecida pelo parâmetro fornecido na busca: “gato em uma caixa”. Este cenário é ilustrado pela Figura 7 abaixo:

Figura 7 – Motor de busca hipotético



Fonte: adaptado de Carnevali (2022)

Na Figura 7, os números representam o *ranking* de relevância que cada imagem poderia representar como predito pelo sistema de recomendação (aqui, o motor de busca hipotético). Podemos notar que os resultados 2, 4, 5 e 7 são resultados relevantes *de fato*. Os outros resultados *não* são relevantes pois eles mostram gatos *sem caixas*, caixas *sem gatos* ou até mesmo um cão.

Quando avaliamos o desempenho de um sistema de recomendação, comparamos as

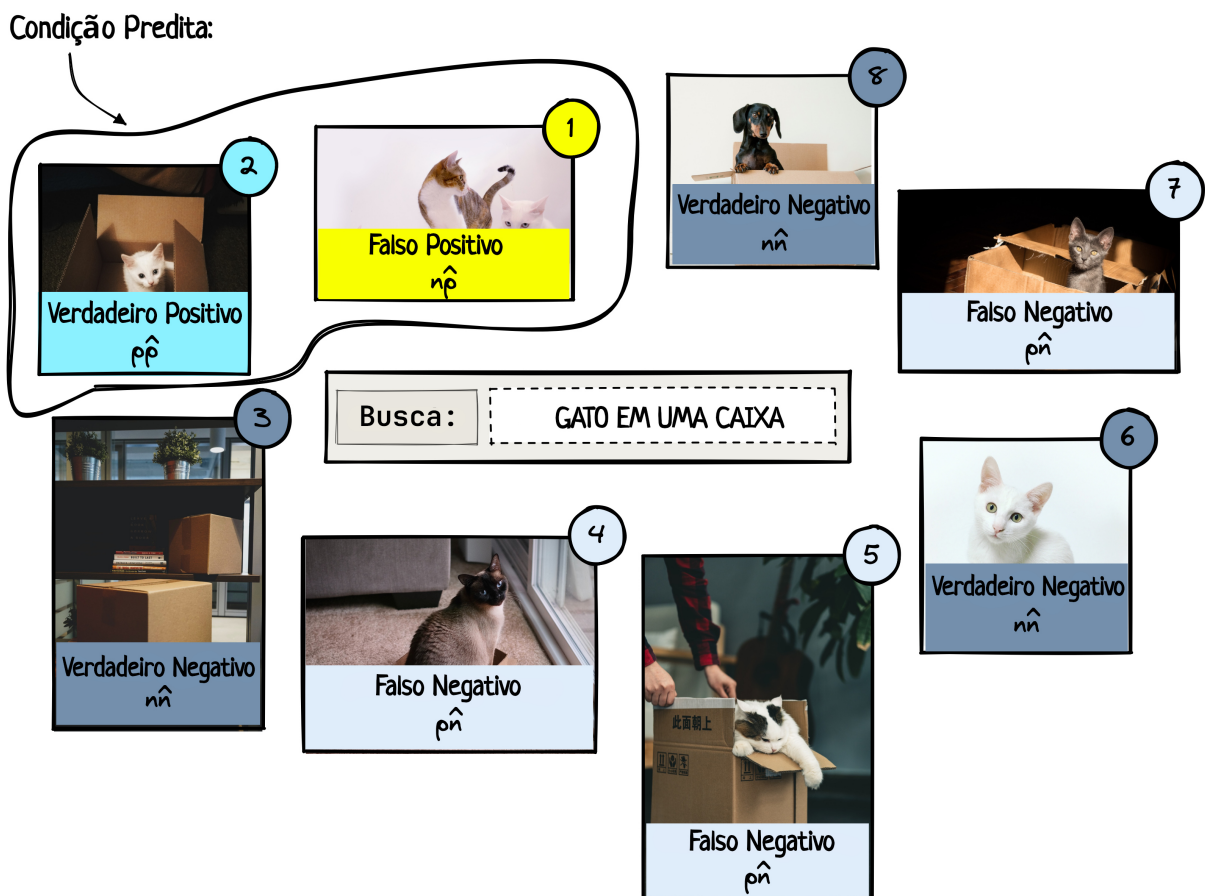
condições *reais* e as *preditas* frente a frente:

- **Condição real:** refere-se ao verdadeiro rótulo de cada item no conjunto de dados. Estes são positivos ( $p$ ) se um item for relevante para uma consulta ou negativo ( $n$ ) se um item for irrelevante para uma consulta.
- **Condição predita:** é o rótulo previsto retornado pelo sistema de recomendação. Se um item for devolvido, é previsto como sendo positivo ( $\hat{p}$ ) e, se não for devolvido, é previsto como negativo ( $\hat{n}$ ).

Os resultados positivos focam no que o sistema de recomendação retorna. Se retornar um resultado real relevante, este é um verdadeiro positivo ( $p\hat{p}$ ); se retornar um resultado irrelevante, temos um falso positivo ( $n\hat{p}$ ).

Para resultados negativos, devemos observar o que o sistema não retorna. Consideremos o caso de retorno de **dois resultados**. Qualquer coisa relevante, mas não retornada, é um falso negativo ( $p\hat{n}$ ). Os itens irrelevantes que não foram devolvidos são verdadeiros negativos ( $n\hat{n}$ ). A Figura 8 exemplifica estas definições com o nosso sistema hipotético de busca:

Figura 8 – Definição de condições sobre a busca



Fonte: adaptado de Carnevali (2022)

## 2.5.1 Métricas Sensíveis à Ordem

### 2.5.1.1 Recall@K

A métrica de *Recall@K*, onde *K* se refere ao número de itens recomendados, é calculada de acordo com a fração de itens relevantes que são recomendados. Ela é definida da seguinte forma:

$$\begin{aligned} \text{Recall@K} &= \frac{\text{verdadeirosPositivos}}{\text{verdadeirosPositivos} + \text{falsosNegativos}} \\ &= \frac{p\hat{p}}{p\hat{p} + p\hat{n}} \end{aligned}$$

A Figura 9 ilustra o cálculo desta métrica utilizando-se do exemplo do motor de busca hipotético. No geral, para valores menores de *K*, a métrica gera valores menores, porém eles tem a tendência a aumentar à medida que o valor de *K* cresce, o que é um ponto negativo do *Recall@K*. O outro ponto negativo, como foi mencionado, é que essa métrica não é capaz de capturar a importância dos itens em relação à ordem de recomendação.

## 2.5.2 Métricas Insensíveis à Ordem

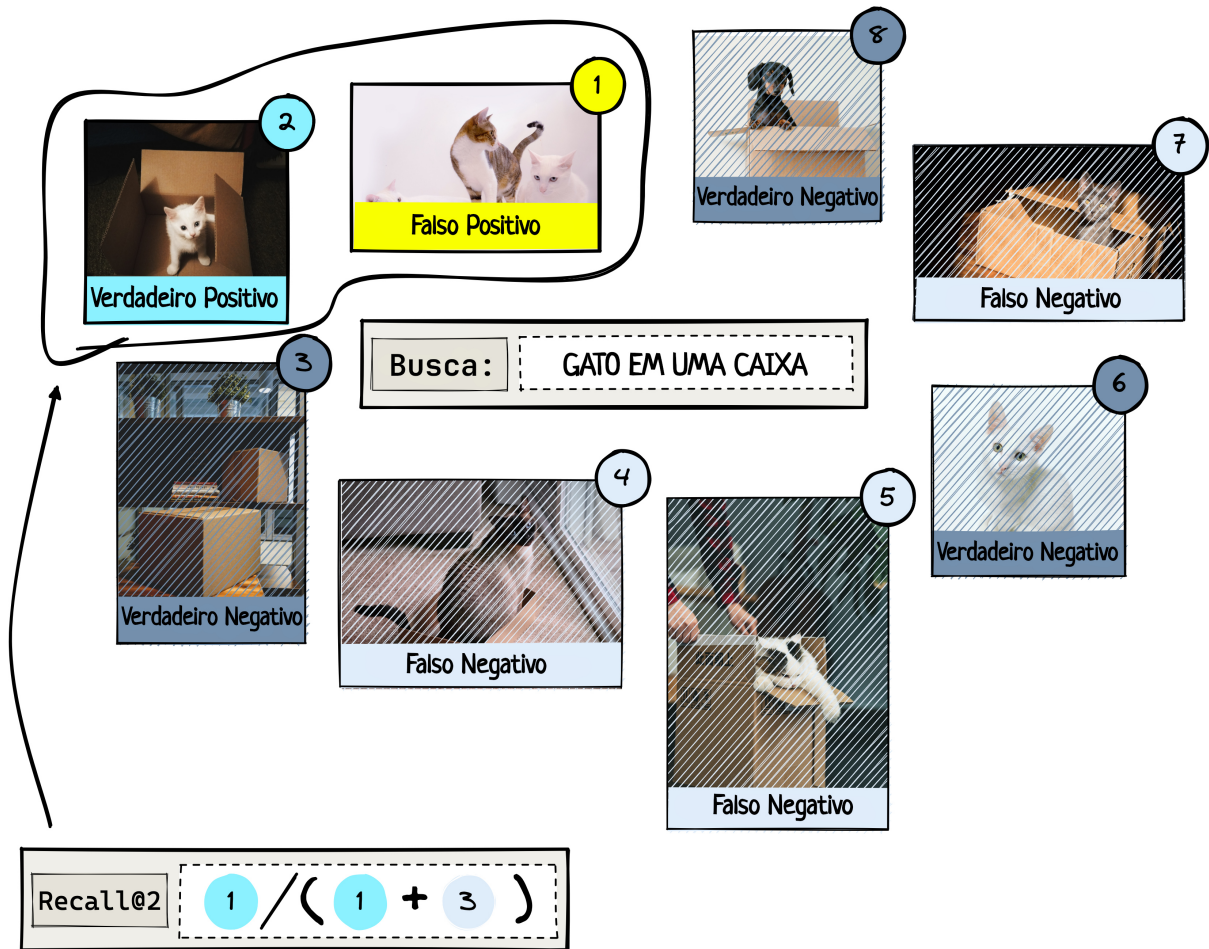
### 2.5.2.1 Mean Reciprocal Rank

Uma das métricas mais simples que tem sensibilidade à ordem, a classificação de significância recíproca, ou *MRR*, é uma medida que usa o primeiro item relevante (e apenas ele) para atribuir essa sensibilidade. É basicamente uma média harmônica dos primeiros índices dos itens relevantes, portanto, é definida da seguinte forma:

$$MRR = \frac{1}{Q} \sum_{q=1}^Q \frac{1}{\text{rank}_q} \quad (2.13)$$

A variável *Q* na definição de *MRR* refere-se ao *número de consultas* realizadas. Isto significa que várias solicitações de recomendação são feitas para que se possa medir a significância que essa métrica busca aferir. A Figura 10 ilustra a execução das buscas com base no sistema de busca proposto anteriormente.

Figura 9 – Cálculo do Recall@K



Fonte: adaptado de Carnevali (2022)

Calculamos o *rank recíproco*  $\frac{1}{rank_q}$  para cada busca/consulta  $q$ . Para a primeira consulta, a primeira imagem relevante real é retornada na posição dois, portanto, a classificação recíproca é  $\frac{1}{2}$ . Ao realizar-se o cálculo da classificação recíproca para todas as consultas:

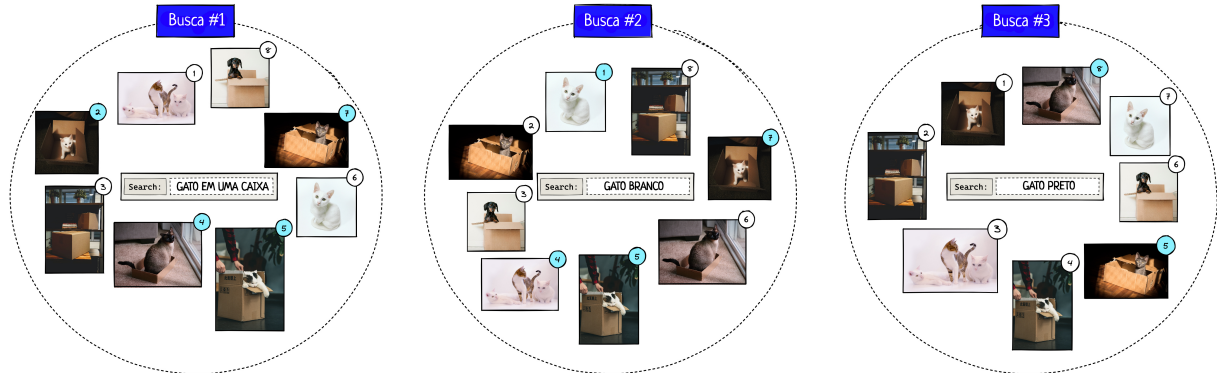
$$busca\ 1 : \frac{1}{rank_1} = \frac{1}{2} = 0,5$$

$$busca\ 2 : \frac{1}{rank_2} = \frac{1}{1} = 1,0$$

$$busca\ 3 : \frac{1}{rank_3} = \frac{1}{5} = 0,2$$

Em seguida, somamos todas essas classificações recíprocas para as consultas  $q = [1, \dots, Q]$  (por exemplo, todas as nossas três buscas):

Figura 10 – Execução das buscas para o cálculo de MRR



Fonte: adaptado de Carnevali (2022)

$$\sum_{q=1}^Q = \frac{1}{rank_q} = 0,5 + 1,0 + 0,2 = 1,7$$

Como estamos calculando a classificação de significância recíproca média (MRR), devemos obter o valor médio dividindo nossas classificações recíprocas totais pelo número de consultas Q:

$$MRR = \frac{1}{Q} \sum_{q=1}^Q = \frac{1}{rank_q} = \frac{1}{3} \cdot 1,7 \approx 0,57$$

A vantagem da MRR é que é uma métrica bastante simples, e é bem eficiente para se encontrar "o melhor item". A desvantagem é que ela foca em apenas um item, desconsiderando o resto da lista de recomendação.

#### 2.5.2.2 Mean Average Precision@k ou Média das Precisões Médias em k

A média das precisões médias em  $k$  ou **MAP@k** avalia se os itens preditos são relevantes, ao mesmo tempo que se preocupa se eles estão no topo da lista. Para se construir a MAP@k, precisamos primeiro definir como calcular a precisão em  $K$ , ou **Precisão@K**, que mede o percentual de itens recomendados que são de fato relevantes. Ela é definida da seguinte forma:

$$Precisão@K = \frac{\text{verdadeirosPositivos}}{\text{verdadeirosPositivos} + \text{falsosPositivos}} = \frac{p\hat{p}}{p\hat{p} + n\hat{p}} = \frac{p\hat{p}}{K} \quad (2.14)$$

Desta forma, podemos definir a precisão média em  $K$ , ou **AP@K**, que calcula a média para todos os valores de  $K$ . Ela é dada por:

$$AP@K = \frac{\sum_{k=1}^K (Precisão@K \cdot rel_k)}{\# \text{ de resultados relevantes}} \quad (2.15)$$

Com essas duas definições, estamos aptos a destacar aqui a definição de MAP@k, que é determinada como:

$$MAP@k = \frac{1}{Q} \sum_{q=1}^Q AP@k_q \quad (2.16)$$

Mais uma vez,  $Q$  se refere ao número de vezes onde recomendações são criadas, as chamadas *consultas*. Para exemplificar o cálculo da MAP@k, podemos retornar ao cenário ilustrado pela Figura 9. Desta forma, de acordo com a definição dada pela Equação 2.14, temos que

$$Precisão@2 = \frac{1}{1+1} = 0,5$$

Segundo a definição dada pela Equação 2.15, precisamos calcular a precisão média em  $k$  para todos os valores de  $k = [1, \dots, K]$ . O fator  $rel_k$  é um parâmetro de relevância que (para  $AP@K$ ) é igual a 1 quando o  $k$ -ésimo item é relevante e é 0 quando não é. Isto significa que toda vez que o item em questão for irrelevante seu valor não é levado em consideração para o cálculo da  $AP@k$ , ou seja, o valor é anulado. Assim, levando-se em conta o cenário proposto pela Figura 9 e as três consultas/buscas propostas pela Figura 10, temos que:

Tabela 1 – Busca #1

Posição $k$	2	4	5	7
$Precisão@k$	0,5	0,5	0,6	0,57
$rel_k$	1	1	1	1

Fonte: elaborada pelo autor.

Tabela 2 – Busca #2

Posição $k$	1	4	5	7
$Precisão@k$	1	0,5	0,6	0,57
$rel_k$	1	1	1	1

Fonte: elaborada pelo autor.

Tabela 3 – Busca #3

Posição $k$	5	8
$Precisão@k$	0,2	0,25
$rel_k$	1	1

Fonte: elaborada pelo autor.

As Tabelas 1, 2 e 3 registram o cálculo de  $Precisão@k$  e  $rel_k$  para todos os itens relevantes ao longo de todas as consultas  $q = [1, \dots, Q]$ . Dados estes valores, podemos calcular o valor de  $AP@K_q$  onde  $K = 8$  da seguinte forma:

$$AP@8_1 = \frac{0,5 \cdot 1 + 0,5 \cdot 1 + 0,6 \cdot 1 + 0,57 \cdot 1}{4} = 0,54$$

$$AP@8_2 = \frac{1 \cdot 1 + 0,5 \cdot 1 + 0,4 \cdot 1 + 0,43 \cdot 1}{4} = 0,67$$

$$AP@8_3 = \frac{0,2 \cdot 1 + 0,25 \cdot 1}{2} = 0,22$$

Cada um dos cálculos de  $AP@K_q$  individuais produz um único valor de  $AP@K$  para cada consulta  $q$ . Finalmente, para obter a  $MAP@K$  para todas as consultas, simplesmente dividimos pelo número de consultas  $Q$ :

$$MAP@K = \frac{1}{Q} \sum_{q=1}^Q AP@k_q = \frac{1}{3} \cdot (0,54 + 0,67 + 0,22) = 0,48$$

Essa métrica tem a vantagem de lidar com ranqueamento de listas naturalmente, em contraste às métricas insensíveis à ordem, que tratam os itens recomendados como conjuntos. Ela também é capaz de penalizar mais fortemente erros que ocorrem na parte de cima das listas recomendadas, porém erros na parte de baixo possuem baixa penalização.

Apesar da  $MAP@k$  ser muito boa para o critério de relevância binário (relevante / não-relevante), ela falha para critérios que possuem degraus de relevância, portanto, níveis intermediários. Ela é incapaz de recuperar a medida da falha deste tipo de informação.

## 2.6 Considerações Finais do Capítulo

Neste capítulo apresentamos os conceitos importantes para o entendimento deste trabalho. Destacamos a desinformação, seus mecanismos e os desafios que eles trazem para a

sociedade como um todo. Estabelecemos as bases lógico-matemáticas de interpretação computacional da linguagem humana, através da vetorização. Enumeramos todos os aspectos relevantes de um sistema de recomendação e como poderíamos ser capazes de avaliar seu desempenho de acordo com o critério necessário para este trabalho. A partir de agora, pretendemos utilizar estas bases de conhecimento para aplicá-lo à problemática levantada pela análise do conjunto de dados alvo deste trabalho, que será apresentada no capítulo seguinte.

### 3 TRABALHOS RELACIONADOS

Nesta seção, será apresentada uma breve revisão dos trabalhos relacionados ao uso de sistemas de recomendação no contexto da desinformação. É notável que os dois elementos (SRs e desinformação), envolvidos em diferentes contextos, podem se relacionar de maneiras distintas. Por exemplo, um SR poderia ser ele próprio um agente difusor de desinformação, considerando todos os mecanismos de propagação de desinformação descritos na Seção 2.1.2, porém há também a possibilidade de ser trabalhado um sistema de recomendação que aja diretamente no *combate* à desinformação.

Exploramos estudos que examinam como os algoritmos de recomendação podem amplificar a disseminação de informações falsas e criar bolhas de informação. Também discutimos iniciativas para melhorar os sistemas de recomendação, como a incorporação de verificação de fatos e diversificação dos resultados recomendados. Também listamos exemplos de sistemas de recomendação que atuam *diretamente* no processo de sugestão de notícias factuais. Consideraremos, doravante, estas diferentes perspectivas ao listar os trabalhos relacionados.

#### 3.1 Contextualização

Existem atualmente poucos trabalhos que buscam investigar o potencial de atuação *direta* de sistemas de recomendação no contexto de disseminação de desinformação. Na ausência de mecanismos eficazes para verificar a veracidade das informações, os sistemas de recomendação podem amplificar essa disseminação. Se um usuário interage com conteúdo desinformativo, o algoritmo pode interpretar isso como um sinal de interesse e, em seguida, recomendar mais conteúdo semelhante. Isso pode criar um ciclo vicioso, onde os usuários são expostos repetidamente a informações falsas, sem uma contraparte factual.

Especificamente, o trabalho de Reis *et al.* (2019) busca investigar o impacto da disseminação de desinformação no contexto do sistema de recomendação de vídeos do YouTube<sup>1</sup>, popular plataforma de compartilhamento e *streaming*. O estudo traz apontamentos sobre os modos de ação desta plataforma no cenário político eleitoral nacional, a atividade dos algoritmos e as estratégias dos atores – sejam eles humanos ou robôs. Neste trabalho, tem-se o intuito de compreender os impactos do sistema de recomendação de vídeos do YouTube, tomando como premissa que os resultados obtidos a cada experiência de busca realizada na plataforma decorrem

---

<sup>1</sup> <https://www.youtube.com>

de um modo singular de comportamento e utilização da plataforma e dos seus usuários que resulta da convergência entre a ação humana e não humana.

O trabalho de Stitini *et al.* (2022) afirma que muitos pesquisadores na área em questão focam na detecção de notícias falsas em si, mas poucos trabalhos destacam o papel dessa detecção na melhoria da qualidade da confiança nos sistemas de recomendação de redes sociais. Nele, é fornecida uma visão geral das abordagens utilizadas para detectar notícias falsas dentro das redes sociais. Tem-se uma apresentação de uma estratégia multiclasse e semissupervisionada de treinamento automático, baseada em um pequeno conjunto de dados classificados e uma grande quantidade de dados não rotulados, para melhorar a precisão de sistemas de recomendação orientados à confiança. A eficiência do método semissupervisionado proposto foi testada em dois conjuntos de dados de referência em relação à precisão da classificação usando os algoritmos de aprendizado simples mais comumente disponíveis: regressão logística, árvore de decisão, Naive Bayes e SVM linear. No experimento realizado, a regressão logística é o modelo que alcança a melhor precisão, com uma pontuação de 96%. Os autores concluem que a abordagem contribui para modelos preditivos multiclasse de detecção de notícias falsas mais eficazes, confiáveis e robustos.

O trabalho de Patankar *et al.* (2019) busca trazer a construção de um sistema de recomendação capaz de identificar viés sociopolítico: os autores afirmam que as soluções atuais concentram-se em estratégias *descendentes*, baseadas em servidores, para decidir se um artigo de notícias é falso ou tendencioso, exibindo apenas notícias confiáveis para os usuários finais. Neste trabalho, utiliza-se uma abordagem de conscientização dos usuários em tempo real sobre a tendenciosidade dos artigos de notícias que eles estão navegando, para que se possa recomendar artigos de notícias de outras fontes sobre o mesmo tópico, com diferentes níveis de tendenciosidade. Com esta pesquisa, os autores esperavam que um sistema com essas características pudesse ajudar os usuários a terem mais consciência do viés presente nos artigos de notícias que estão lendo no momento.

A pesquisa desenvolvida por Sallami *et al.* (2023) age no âmbito de mitigação de *notícias falsas* através da aplicação direta de sistemas de recomendações. O trabalho apresenta a construção de um sistema de recomendação sensível a notícias falsas (FANAR, de acordo com os autores). Trata-se de uma adaptação do algoritmo de filtragem colaborativa que modela usuários online não confiáveis para removê-los da vizinhança do usuário candidato. O trabalho dos autores também propõe a criação do FNEWR, um conjunto de dados para o sistema de

recomendação desenvolvido em questão. Os experimentos desenvolvidos revelam que o FANAR supera as principais técnicas de recomendação de notícias atuais em suas capacidades de sugerir notícias personalizadas e mitigar a disseminação de informações falsas.

Por fim, o trabalho de Wang *et al.* (2022) propõe um novo modelo de recomendação baseado em veracidade e orientado a eventos para recomendar notícias verdadeiras corretivas personalizadas a usuários individuais para desmascarar efetivamente notícias falsas, o *Rec4Mit*. Segundo os autores, o sistema não apenas captura de forma eficaz as preferências de leitura atuais do usuário, mas também prevê com precisão a veracidade (verdadeira ou falsa) das notícias candidatas. Dessa forma, o *Rec4Mit* pode recomendar as notícias verdadeiras mais adequadas para melhor atender às preferências do usuário e mitigar as notícias falsas. Experimentos desenvolvidos pelos autores em conjuntos de dados do mundo real mostram que o *Rec4Mit* supera significativamente os métodos de recomendação de notícias do estado da arte em termos da capacidade de recomendar notícias verdadeiras personalizadas para mitigação de notícias falsas.

### **3.2 Considerações Finais do Capítulo**

Nesta seção, realizamos uma breve análise descritiva dos trabalhos relacionados encontrados na literatura que estabelecem a relação entre sistemas de recomendação e desinformação. Os trabalhos de Stitini *et al.* (2022) e Sallami *et al.* (2023) trazem a tona a existência de muitas pesquisas que focam na detecção de desinformação propriamente dita (por meio da identificação de notícias falsas), mas sugerem que poucos trabalham a melhoria da qualidade dos *próprios* sistemas de recomendação que atuam nas redes sociais. O trabalho de Wang *et al.* (2022) traz uma ideia interessante de sugestão de informação, podendo detectar sua qualidade de conteúdo no que concerne à desinformação.

O trabalho atual desta pesquisa pretende desenvolver uma ideia relativamente semelhante à de Wang *et al.* (2022), porém voltada para filtragem de dados no WhatsApp no contexto de compartilhamento de desinformação no Brasil, que é algo que ainda é pouco evidente no campo de estudos da área. No capítulo seguinte será discutido com detalhes o processo de criação e de análise exploratória do conjunto de dados utilizados para a construção do sistema de recomendação proposto pelo trabalho, considerando o contexto de pesquisa mencionado anteriormente.

## 4 FAKEWHATSAPP.BR

Neste capítulo será feita uma breve análise descritiva de todas as etapas que foram necessárias para a coleta, o processamento e a análise exploratória dos dados que serão utilizados neste trabalho. Os dados podem ser encontrados no repositório online<sup>2</sup>.

### 4.1 Contexto e Razões

Poucos são os trabalhos na língua portuguesa que trabalham o conceito de desinformação no aspecto do desenvolvimento de métodos de detecção e identificação automática de elementos desinformadores no WhatsApp, especialmente no contexto de sistemas recomendativos. Existem alguns deles que exploram a tática de desenvolvimento de um conjunto de dados coletados diretamente do WhatsApp e de outras plataformas, de acordo com metodologias e critérios próprios, para que sirvam de base ao desenvolvimento das análises implementadas sobre este contexto.

Em particular, queremos referenciar três deles: o trabalho de Monteiro *et al.* (2018) foi um dos primeiros a ofertar a criação de um corpus para análise de notícias falsas em língua portuguesa. Nele foi proposto a criação do corpus Fake.Br que contempla 7200 notícias, classificadas como verdadeiras ou falsas, que foram selecionadas dentre diversos sites de notícias e se propunha a ser um conjunto de dados de referência para testes de algoritmos de identificação de notícias falsas em artigos publicados na Web através de portais, blogs e veículos de comunicação online. O trabalho de Cordeiro e Pinheiro (2019) teve uma inspiração neste processo de criação de corpus com essas características e propôs uma ideia similar, entretanto, buscou construir um corpus voltado para a coleta de conteúdo textual da plataforma de *microblogging* Twitter<sup>3</sup> chamado FakeTweet.Br. Por fim, (CABRAL, 2021) e (CABRAL *et al.*, 2021) evidenciam o processo de criação de um corpus semelhante, o FakeWhatsApp.BR, para a plataforma WhatsApp. Ele foi desenvolvido no contexto específico das eleições presidenciais brasileiras de 2018, que consta um rótulo (atributo) característico para desinformação.

O conjunto de dados alvo deste trabalho é basicamente uma variante do FakeWhatsApp.BR criada em 2020 que segue as mesmas diretrizes de coleta e rotulação, entretanto possui o diferencial de que o contexto das mensagens obtidas é referente à pandemia de COVID-19. Iremos doravante referenciá-lo como ele é definido no repositório, com o nome `misinformation_covid19`.

<sup>2</sup> [https://gitlab.com/jmmonteiro/misinformation\\_covid19](https://gitlab.com/jmmonteiro/misinformation_covid19)

<sup>3</sup> <https://twitter.com>

## 4.2 Estratégia de Criação do *Corpus*

A plataforma de mensagens WhatsApp não disponibiliza uma API pública para coleta de dados, exigindo que métodos alternativos sejam empregados. Assim como foi declarado anteriormente, a estratégia de coleta aqui demonstrada funciona da mesma maneira trabalhada na construção do corpus *FakeWhatsApp.BR*, portanto apenas o contexto de busca do conteúdo será diferente, que envolverá a temática de desinformação referente de maneira direta ou indireta à pandemia de SARS-CoV-2.

Para isto, de acordo com (CABRAL, 2021) a estratégia de geração dos dados é dada em três etapas principais:

1. Coleta dos dados brutos;
2. Normalização;
3. Rotulação.

Estes passos serão evidenciados de agora em diante para que sejam demonstrados detalhes intrínsecos à metodologia empregada.

### 4.2.1 *Coleta de Dados Brutos*

Uma vez que não existem mecanismos mais simples e/ou transparentes como outras plataformas como o Twitter ou o Telegram<sup>4</sup> (uma API que possa ser consumida para viabilizar a coleta massiva de dados, por exemplo), este método busca interagir com a plataforma de maneira semi-automatizada, isto é, o processo de recepção dos dados é feita, em parte, ao se simular um agente humano que é capaz de entrar em certos grupos, de acordo com certos critérios e, que poderia, em tese, interagir normalmente como qualquer integrante (porém não o faz porque isto não interessa à coleta) e, por outro lado, pelo uso de mecanismos disponibilizados pela plataforma que permitem que este suposto *agente humano* possa exportar os dados de suas conversas a partir de seus grupos.

Desta forma, uma conta é criada no aplicativo. Esta conta deverá gozar de todas as capacidades e possibilidades de um usuário comum, portanto poderia, em tese, conversar com outros prováveis contatos, fazer ligações, bloquear números, entrar em grupos e visualizá-los etc. Entretanto, para este passo apenas a última ação será relevante. É importante ressaltar que, como este agente está sendo ingressado como um usuário comum, esta conta está sujeita e restrita

---

<sup>4</sup> <https://telegram.org>

aos termos de uso e condições de acesso do WhatsApp e, para que estas circunstâncias sejam obedecidas, alguns tratamentos mitigatórios serão evidenciados mais à frente.

Esta conta é então utilizada para que seja possível entrar em grupos definidos com visibilidade pública, relacionados a temas políticos e afins, através da palavra chave "chat.whatsapp.com". A partir dos primeiros grupos, torna-se relativamente fácil de se encontrar links e referências para outros grupos de caráter similar. A conta permanece nesse grupos em um período fixo pré-estabelecido de tempo, sem realizar interações. Após o término deste período, as mensagens são extraídas utilizando-se um recurso da plataforma que permite exportar todas as mensagens na forma de um arquivo de texto plano. As mensagens são armazenadas em linhas de texto, onde cada linha inclui os seguintes dados, nesta ordem: a data, a hora, o número do celular do autor e o texto em si.

O conjunto de dados brutos, após esse procedimento de exportação, deverá gerar  $n$  arquivos de texto, onde  $n$  representa o número de grupos, portanto, cada grupo é capaz de exportar um arquivo. Devido a limitações da plataforma, apenas conteúdo de tipo textual poderá ser extraído dos grupos. Áudios, imagens e vídeos não são incluídos no conjunto de dados. Quando um arquivo de mídia com essas características é enviado, a descrição genérica <Arquivo de mídia oculto> é gerada em seu lugar.

#### 4.2.2 Normalização dos Dados

Após o término do processo de coleta, o próximo passo consiste em realizar um processamento de todos os arquivos de texto e implementar um reprocessamento desses dados em uma estrutura de tabela. Isto é feito da seguinte maneira: todos os arquivos são lidos programaticamente e seu conteúdo textual é condensado em um único local, separado por cada grupo. A partir do uso de expressões regulares, os atributos dentro de cada cadeia de caracteres são identificados, de acordo com o padrão <data><hora><telefone><mensagem>. Para a estrutura de tabela final, cada linha é a representação dos dados de uma única mensagem, com cada coluna representando todos seus atributos e a mensagem propriamente dita. Mensagens automáticas geradas pelo aplicativo são descartadas (por exemplo, "usuário saiu do grupo", "nome do grupo foi alterada"etc). Ao término deste reprocessamento, um arquivo do tipo *valores separados por vírgula* (CSV) é criado.

Para que a privacidade dos usuários seja preservada, existe uma etapa de normalização que busca **anonimizar** as mensagens. Isto é realizado por meio de uma função *hash*

que mapeia os números de telefones para um identificador único e substitui este número de telefone por seu respectivo identificador na tabela. Outros elementos textuais indesejados dentro do conteúdo das mensagens em si, como números de telefone, são removidos. Esta etapa garante que tanto os requerimentos da política de privacidade<sup>5</sup> da plataforma bem como os da Lei Geral de Proteção de Dados Pessoais (LGPD)<sup>6</sup> sejam atendidos. Um atributo que identifica o tipo de conteúdo da mensagem (texto ou mídia) também é adicionado. Por fim, é estabelecido um critério para contabilizar a presença de conteúdo dito "viral", que consiste em contar o número de ocorrências de mensagens repetidas dentro do conjunto de dados. Mensagens irrelevantes (cumprimentos e respostas curtas) são ignoradas pelo critério de contagem.

### 4.2.3 Rotulação das Classes

A última etapa de geração do conjunto de dados, ainda de acordo com o método descrito por (CABRAL, 2021), prevê um processo de **rotulação** manual das mensagens. De acordo com este método, a problemática de detecção de desinformação é trabalhada como um problema de classificação binária, onde as mensagens que constem provável desinformação são atribuídas a classe positiva para o rótulo *misinformation*, enquanto que as "sem desinformação" são atribuídas a classe negativa para este rótulo. Uma vez que o processo de rotulagem manual das mensagens é cansativo e demorado, a estratégia é escolher um subconjunto das mensagens que seja interessante a atribuição dos rótulos. O critério escolhido por (CABRAL, 2021) é o de utilizar apenas as mensagens ditas *virais*, aquelas que tem na sua elaboração o intuito de serem compartilhadas e espalhadas diversas vezes. Existem algumas diretrizes definidas por (CABRAL, 2021) para esta tarefa de rotulação. Elas estão definidas pelo fluxograma descrito pela Figura 11.

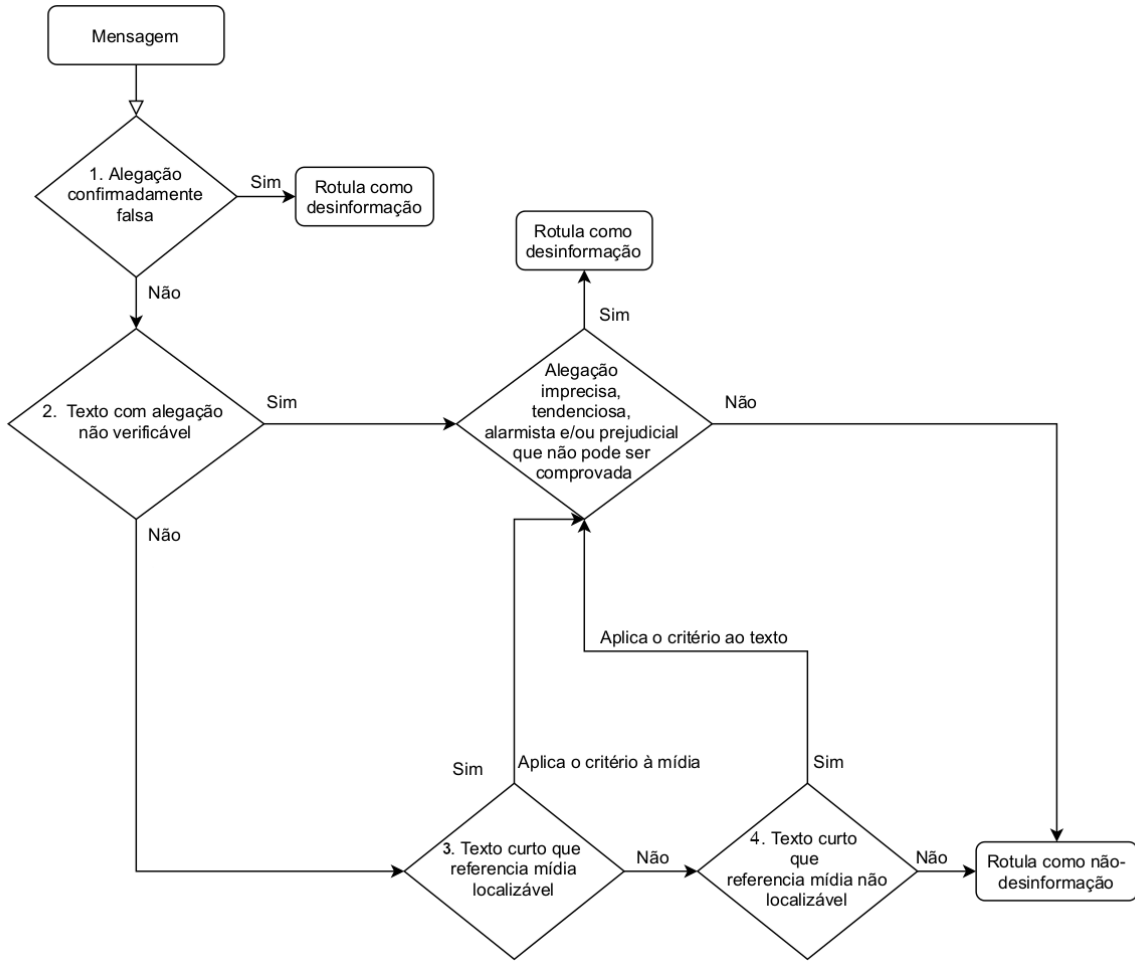
Para méritos de validação de veracidade, as informações são cruzadas com fontes encontradas em grandes plataformas de jornalismo e outras plataformas brasileiras de checagem de fatos. Após a tarefa de rotulação manual, o subconjunto rotulado de mensagens é incorporado ao original, e as mensagens não contempladas permanecem sem rótulo.

A partir do momento que as três etapas anteriores são concluídas, o processo de geração de dados está finalizado. Em suma, um conjunto maior de dados brutos é resultante do processo de normalização. Temos um subconjunto ainda menor após o recém-mencionado

<sup>5</sup> [https://www.whatsapp.com/legal/privacy-policy/?lang=pt\\_br](https://www.whatsapp.com/legal/privacy-policy/?lang=pt_br)

<sup>6</sup> <https://www.gov.br/cidadania/pt-br/acesso-a-informacao/lgpd>

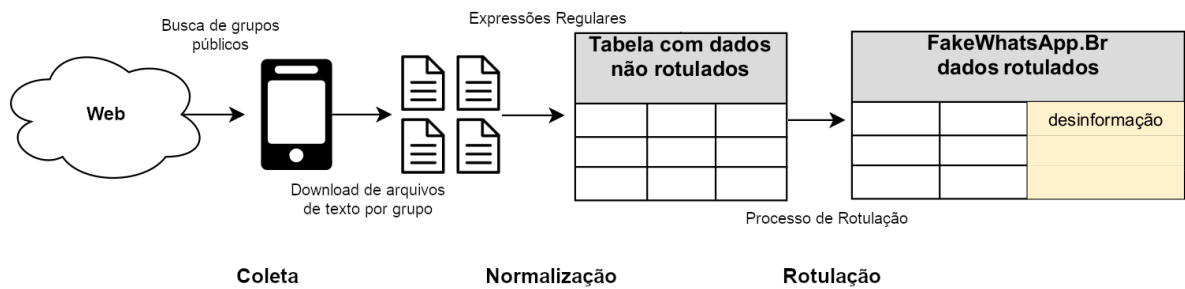
Figura 11 – Diretrizes de rotulação das mensagens



Fonte: adaptado de Cabral (2021)

processo de rotulação. A Figura 12 mostra uma visão de alto nível do fluxo de todo o processo.

Figura 12 – Síntese do processo de geração de dados



Fonte: adaptado de Cabral (2021)

### 4.3 Análise Exploratória

O conjunto de dados resultante deste processo de geração de dados será o objeto de estudo desta pesquisa, que pode ser encontrado na sua versão do ano de 2020, em seu repositório<sup>7</sup>. Ele possui 2899 mensagens, definidas por cinco atributos: o **texto** em si, que define a própria mensagem, o **número de compartilhamentos**, que representa a quantidade que aquela mensagem se repetiu no conjunto original de dados, uma coluna de **fonte**, que mostra uma referência utilizada para o critério de checagem manual e rotulação, a coluna que define o rótulo de **desinformação**, que indica 0 para ausência de conteúdo desinformativo, 1 para presença de algum conteúdo desinformativo, e nulo como ainda não classificado. Por fim, há uma coluna de **revisão**, que determina que aquele registro foi validado mais de uma vez.

Todo o pré-processamento de dados, visualizações e algoritmos desenvolvidos no presente trabalho foram desenvolvidos na linguagem de programação Python<sup>8</sup>, construídos sob o leque de ferramentas providas pelas seguintes bibliotecas:

- a) numpy<sup>9</sup> para cálculos matemáticos;
- b) pandas<sup>10</sup> para manipulação de dados tabulares;
- c) matplotlib<sup>11</sup> e seaborn<sup>12</sup> para gráficos e visualizações;
- d) scikit-learn<sup>13</sup> para tarefas preditivas, cálculo de métricas e uso de aprendizado de máquina.

A Figura 13 representa a visão da amostra dos dados após o processo de criação. Essa é uma pré-visualização amigável dos dados CSV fornecidos à biblioteca pandas, com todos os seus atributos.

Figura 13 – Visão do conjunto de dados

	shares	text	misinformation	source	revision
0	27	O ministro da Ciência, Tecnologia, Inovações e...	0.0	<a href="https://www.gov.br/pt-br/noticias/educacao-e-p...">https://www.gov.br/pt-br/noticias/educacao-e-p...</a>	NaN
1	26	Pesquisa com mais de 6.000 médicos em 30 paíse...	1.0	<a href="https://www.aosfatos.org/noticias/e-falso-que-...">https://www.aosfatos.org/noticias/e-falso-que-...</a>	NaN
2	25	É com muita alegria que comunico que mais um p...	0.0	<a href="http://portal.mec.gov.br/component/content/art...">http://portal.mec.gov.br/component/content/art...</a>	NaN
3	25	Renda Brasil unificará vários programas social...	0.0	<a href="https://agenciabrasil.ebc.com.br/politica/noti...">https://agenciabrasil.ebc.com.br/politica/noti...</a>	NaN
4	24	O Secretário-Geral da OTAN Jens Stoltenberg ta...	0.0		NaN 1.0

Fonte: o autor.

<sup>7</sup> [https://gitlab.com/jmmonteiro/misinformation\\_covid19/-/tree/main/data/2020](https://gitlab.com/jmmonteiro/misinformation_covid19/-/tree/main/data/2020)

<sup>8</sup> <https://www.python.org>

<sup>9</sup> <https://numpy.org>

<sup>10</sup> <https://pandas.pydata.org>

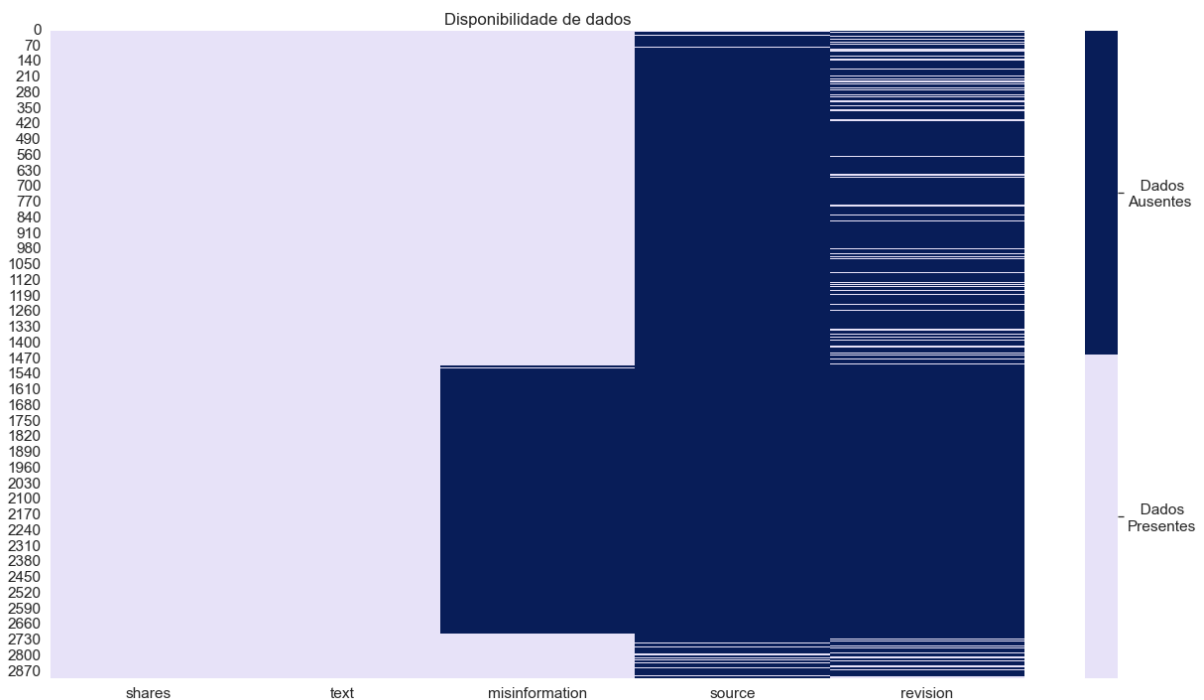
<sup>11</sup> <https://matplotlib.org>

<sup>12</sup> <https://seaborn.pydata.org>

<sup>13</sup> <https://scikit-learn.org>

Ao analisarmos a presença dos dados, vemos que existem algumas colunas com grandes lacunas. Limitações de tempo, coleta e pesquisa podem ter impactado na completude destes atributos. Entretanto, os atributos mais importantes (texto, fonte e rótulo de desinformação) ainda estão bastante presentes. Existe uma grande lacuna para os dados rotulados dentro deste conjunto de dados, mas ainda assim existe uma quantidade substancial de mensagens para que se possa desenvolver uma análise satisfatória. A Figura 14 traz uma visualização dessas lacunas ao longo das colunas do conjunto de dados. Se considermos apenas o atributo `misinformation`, que indica a distribuição de dados rotulados como desinformação, temos o cenário apontado pela Figura 15.

Figura 14 – Visualização de dados faltantes

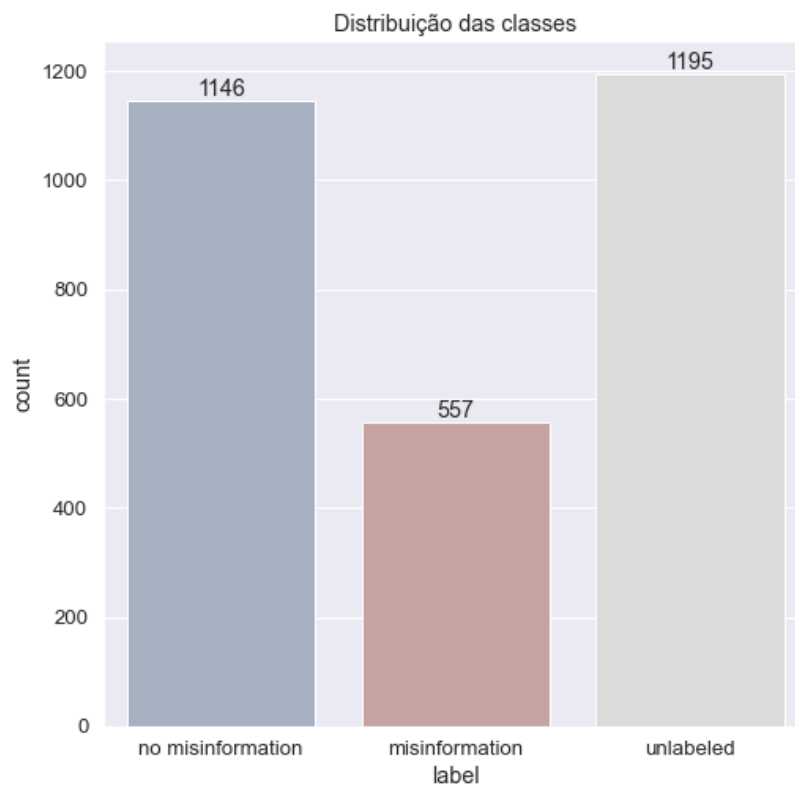


Fonte: o autor.

#### 4.3.1 Extração de Informação

O atributo mais importante e que contém maior quantidade de informação é o próprio texto das mensagens. Entretanto, da maneira que está representado agora, o conjunto de dados aparenta trazer pouca informação, mas a verdade é que esta informação está *condensada* dentro deste atributo. Podemos aprofundar a análise de todos esses textos e extrair informações valiosas deles se aplicarmos um processo denominado *engenharia de atributos* que, para este caso, nada mais consiste em um processo de criação de novos atributos derivados do original visando a

Figura 15 – Distribuição das classes



Fonte: o autor.

extração de mais informações (NARGESIAN *et al.*, 2017).

A partir deste atributo, podemos derivar informações como contagem de palavras, presença de caracteres especiais, contagem de *emojis*, contagem de *hashtags*, que muitas vezes trazem informações mais representativas de maneira resumida, entre outras características. Por fim, poderemos também, em um novo atributo, remover palavras que geralmente não adicionam informação ao conteúdo por serem redundantes - as chamadas **palavras vazias**, ou *stopwords*, de forma a deixar o conteúdo textual mais limpo para um eventual processamento semântico do conteúdo.

Por meio da definição de funções auxiliares no código, considerando o conteúdo textual da mensagens, criamos vários novos atributos aplicando as seguintes operações de contagem:

- a) Número de caracteres;
- b) Número de palavras;
- c) Número de letras maiúsculas;
- d) Número de palavras maiúsculas;

- e) Número de pontuações;
- f) Número de palavras em citações;
- g) Número de sentenças;
- h) Número de palavras únicas;
- i) Número de *hashtags*;
- j) Número de menções (@) no texto;
- k) Número de palavras vazias;
- l) Número de *links*.

Também foi criada uma nova coluna de texto "limpo" que aplica as seguintes operações de "limpeza":

1. Remoção de links;
2. Remoção de palavras em branco;
3. Imposição de letras minúsculas;
4. Remoção de termos de risada.

Essas operações puderam ser realizadas com métodos simples de manipulação de cadeias de caracteres e com o uso de expressões regulares. Para a contagem de sentenças, foi aplicado o conceito de *tokenização*, que separa a mensagem em partes de sentenças menores. Boa parte desses processos foram implementados com aplicações da biblioteca `nltk`<sup>14</sup>, amplamente utilizada na área de Processamento de Linguagem Natural.

#### 4.3.2 *Visualização de Frequência de palavras*

Uma vez que temos esses dados divididos e categorizados em seus atributos, podemos explorar alguns conceitos estatísticos quanto a essas características. Se agruparmos os dados por rótulo, poderemos ter uma boa perspectiva de seu comportamento quanto a tendências e correlações estatísticas, tendo em vista o encontro de algum padrão que caracterize a desinformação. Realizamos este processo de agrupamento e calculamos, para cada um dos atributos numéricos, as seguintes métricas estatísticas: *média*, *desvio-padrão*, *mínimo*, *quartis acumulados* (25%, 50%, 75%) e o *máximo*.

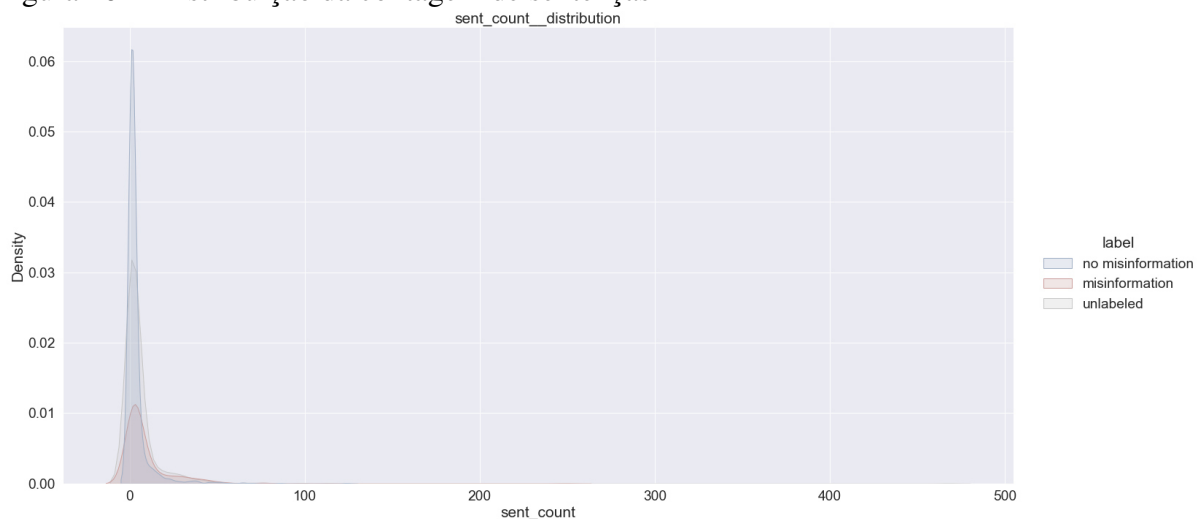
Alguns desses atributos apresentaram irregularidades no cálculo dessas métricas por motivo de falta de dados ou mesmo porque os dados disponíveis eram insuficientes. Por exemplo, a contagem de mensagens com palavras entre aspas indicou zero para algumas das ocorrências,

<sup>14</sup> <https://www.nltk.org>

resultando em um desvio padrão nulo. Entretanto, algumas dessas métricas trouxeram algumas percepções interessantes. Para traçarmos uma análise qualitativa e quantitativa, serão demonstrados os dados de dois atributos em específico de duas maneiras: (I) a representação aproximada da distribuição, que usa o método de *estimativa de densidade por kernel* (ROSENBLATT, 1956) (PARZEN, 1962) e (II) os valores absolutos das métricas estatísticas mencionadas anteriormente expostos de maneira explícita.

A Figura 16 e a Tabela 4 apresentam respectivamente as distribuições e as métricas estatísticas para o atributo que registra a **contagem de sentenças das mensagens**, agrupadas por rótulo. Essa visualização da densidade nos indica que mensagens desinformativas tem a *tendência* a possuírem *mais* sentenças, pelo menos neste conjunto de dados. Os valores dos últimos quartis (75% e máximo) maiores para desinformação corroboram para isto, bem como o fato de que a área vermelha do gráfico localizada à direita do pico aparenta ser maior para a distribuição vermelha (desinformação). Em outras palavras, a *probabilidade* de uma mensagem ser classificada como desinformação é *maior* para valores maiores de sentenças.

Figura 16 – Distribuição da contagem de sentenças



Fonte: o autor.

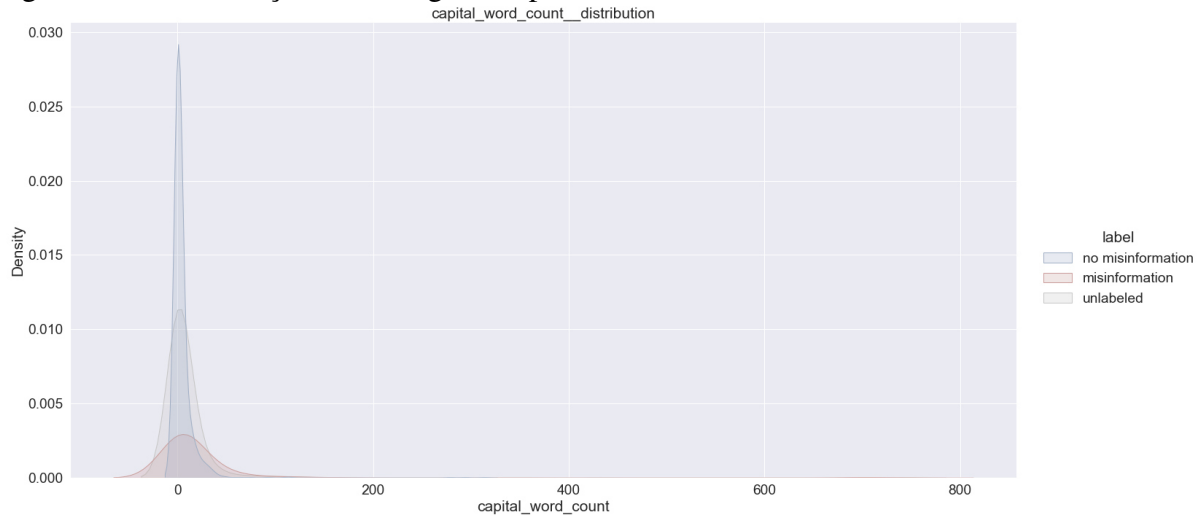
A Figura 17 e a Tabela 5 apresentam a mesma perspectiva, porém agora para a **contagem de palavras com caixa alta**. O mesmo padrão se repete, ou seja, a probabilidade de uma mensagem ter desinformação é maior para valores maiores de palavras com caixa alta. Para se ter uma ideia de como esses dois padrões se apresentam nas mensagens, foram escolhidas aleatoriamente cinco mensagens de cada tipo (com e sem desinformação). As Figuras 18 e 19 exemplificam, respectivamente, essas amostras.

Tabela 4 – Contagem de sentenças

	Desinformação	Sem Desinformação	Não-Rotulado
<b>Média</b>	9.4	3.6	5.7
<b>Desvio-Padrão</b>	17.3	8.3	16.9
<b>Valor Mínimo</b>	1.0	1.0	1.0
<b>Q25</b>	2.0	1.0	1.0
<b>Mediana</b>	3.0	1.0	1.0
<b>Q75</b>	9.0	2.0	4.0
<b>Valor Máximo</b>	249.0	124.0	468.0

Fonte: elaborada pelo autor.

Figura 17 – Distribuição da contagem de palavras com caixa alta



Fonte: o autor.

Tabela 5 – Contagem de palavras com caixa alta

	Desinformação	Sem Desinformação	Não-Rotulado
<b>Média</b>	21.5	4.7	10.2
<b>Desvio-Padrão</b>	77.3	17.6	51.2
<b>Valor Mínimo</b>	0.0	0.0	0.0
<b>Q25</b>	1.0	0.0	0.0
<b>Mediana</b>	3.0	0.0	1.0
<b>Q75</b>	15.0	3.0	5.0
<b>Valor Máximo</b>	748.0	314.0	748.0

Fonte: elaborada pelo autor.

Uma visualização interessante que apresenta a **frequência** das palavras é conhecida como **nuvem de palavras**. Nela, o tamanho das palavras é representado proporcionalmente às suas frequências, ou seja, expressões mais frequentes aparecem maiores na nuvem. É um bom mecanismo visual que demonstra a expressividade de certas palavras (ou expressões) dentro de um *corpus* textual. Geramos essa visualização para as duas classes rotuladas, utilizando o atributo que possui o texto das mensagens tratado (remoção de palavras vazias, links, urls etc). As Figuras 20 e 21 trazem essa representação.

## Figura 18 – Exemplos de mensagens desinformativas escolhidas aleatoriamente

LABEL: misinformation

👉 OLHEM ISSO o CORONE CIRO GOMES TOCANDO O HORROR EM FORTALEZA CEARÁ. ELE É O CANDIDATO DO PARTIDO SOCIALISTA CHINÊS, AQUI NO BRASIL 🇨🇳🇨🇳🇨🇳

LABEL: misinformation

\*Se a ordem do Ditador Comunista-Globalista do DemoDoria é inconstitucional e ilegal POR QUE O POVO QUE VIROU GADO OU TRA VEZ POR CAUSA DA PROPAGANDA HIPNOTICA DA GRANDE MERDIA/EXTREMA IMPRENSALHA SOBRE A FARSA DO VIRUS CHINES NÃO SAI AS RUAS EM MASSA NÃO SAINDO E FICANDO NAS RUAS ATÉ ACAMPANDO NA FRENTE DA ALESP NO PARQUE DO IBIRAPUEIRA EM SINAL DE DESOBEDIÊNCIA CIVIL PRA FORÇAR A ALESP PRA IMPICCHAR ESTE DEMONIO PELO MENOS E SALVAR A ECONOMIA DO BRASIL COMO FIZERAM PRA IMPICCHAR A RAINHA DA MANDIOCA????\*

POR QUE ESSES "CONSERVADORES" QUE DIZEM SER TÃO PATRIOTAS QUE AMAM O BRASIL DE VERDADE NÃO TIRAM DINHEIRO DO SEU PROPRIETÁRIO FAZENDO UMA VAQUINHA PRA LUTAR NA JUSTIÇA CONTRA O DECRETO PORRA???

<https://criticanacional.com.br/2020/04/06/joao-doria-prossegue-no-ataque-a-economia-paulista-e-prorroga-quarentena/>

LABEL: misinformation

@ElmindoP 🇺🇸 PATRIOTA!

\*VAMOS ACORDAR BRASIL!!!! E VOCE AINDA ACREDITANDO NESTA FARSA DE COVID19, OI ARQUITETADO PARA ENGANAR OS BRASILEIROS, MENOS ESCLARECIDOS...\*

É UM GOLPE QUE F

#NAOFIQUEM CASA

#VAMOS TRABALHAR

#BOLSONARO ESTÁ CERTO

<https://youtu.be/qv7c0LBCZfs>

LABEL: misinformation

UNIAO EUROPEIA apoia a INTERVENÇÃO contra o ditador apoiado pelo PT Psol PSDB PCDOB.. **\*\*\*\*\***  
MILHARES DE PESSOAS ESTÃO SENDO ASSASSINADAS PELO DITADOR SOCIALISTA DA Venezuela. Sabe qual é causa mortis que lança m nas estatísticas: ? Virus Chines.

🇺🇸🇺🇸🇺🇸🇺🇸

Parece com alguma coisa tupiniquim

[https://www.em.com.br/app/noticia/internacional/2020/04/03/interna\\_internacional,1135497/ue-apoia-plano-dos-eua-para-governo-de-transicao-na-venezuela.shtml](https://www.em.com.br/app/noticia/internacional/2020/04/03/interna_internacional,1135497/ue-apoia-plano-dos-eua-para-governo-de-transicao-na-venezuela.shtml)

LABEL: misinformation

\*AMEAÇA DO GLOBALISTA/COMUNISTA DO BILL "CHIP DA BESTA" GATES: NINGUEM PODERÁ SAIR MAIS DE SUAS CASAS QUE FORAM CONVERTIDAS EM PRISÕES OU PODER TRABALHAR ATÉ QUE RECEBAM A VACINA VENENOSA DO VIRUS CHINES QUE É UMA FARSA QUE MATA AOS POCOS E COLOCA O CHIP DA BESTA NO POVO\*

\*A Inglaterra, Italia, Texas e outros Desgovernos tem publicamente anunciado o plano diabolico -- Sintonzem e compartilhem este link extraordinario agora!\*

\*Alex Jones ALERTA que os Governantes Tirânicos deste planeta estão preparando diretivas para obrigar os seus cidadãos a serem CHIPADOS E ENVENENADOS VIA VACINAS se quiserem ter a esperança de reentrar na sociedade.\*

\*Os Comunistas/Globalistas estão planejando usar a força pra forçar a inoculação de doenças piores de verdade e graves como a AIDS e o EBOLA + Vigilância Total + Abolição da Privacidade PRA NOS CONTROLAR FEITO ESCRAVOS ESTÁ AQUI!\*

O Bill Gates pensa que ele é o Imperador dos EUA e que manda neste mundo!!!!

<https://www.infowars.com/watch-live-saturday-coronavirus-news-briefing/>

Fonte: o autor.

Ao analisarmos o contexto do conjunto de dados e relacionarmos esta análise com a visualização de nuvem de palavras, percebemos que a prevalência de certos termos tem forte influência no sentido das entidades mencionadas. Por exemplo, as declarações pejorativas e negativas do país de origem da pandemia de COVID-19, são um exemplo recorrente nas amostras de mensagens com desinformação, e sua frequência de uso é evidenciada pela nuvem de palavras correspondente. Já para o caso da nuvem de "informação", apesar de alguns termos estarem presentes também na de "desinformação", no primeiro caso eles são bem mais balanceados, neutros e equilibrados. Por exemplo, a forte presença das palavras "combate",

## Figura 19 – Exemplos de mensagens não-desinformativas escolhidas aleatoriamente

LABEL: no misinformation

\*🚨URGENTE! BOLETIM ESTADO DE MINAS GERAIS DOMINGO - (22/03/2020)🚨\*

\*Informe Epidemiológico Coronavírus\*

\*Até o momento, foram notificados 7.273 casos de infecção humana pelo COVID-19, em 12 municípios de Minas Gerais. Des-tes casos, 7.190 estão em investigação como\* \*suspeitos e oitenta e três (83) casos foram confirmados.\*

\*Foram notificados casos nos seguintes municípios: Belo Horizonte; Coronel Fabriciano; Divinópolis; Ipatinga; Juiz de Fora; Mariana; Nova Lima; Patrocínio; Poços de Caldas; Sete Lagoas; Uberaba e Uberlândia.\*

[https://m.facebook.com/story.php?story\\_fbid=2722579181299414&id=1879630128927661](https://m.facebook.com/story.php?story_fbid=2722579181299414&id=1879630128927661)

LABEL: no misinformation

Marquem os deputados federais no comentário!

A Câmara Federal, na pessoa do " grande defensor da democracia - @rodrigomaiarj ", rejeitou o uso do Fundão para o combate ao corona vírus.

Peço aos Deputados Federais da Paraíba que não deixem o gordinho malasombro emperrar a PL do @marcofeliciano , que re-duzirá pela metade vossos salários para o combate ao corona vírus.

Tenham consciência. O Vereador, Deputado, Senador, qualquer político que resiste em cortar custos da máquina, está contra a população.

Marquem os deputados federais no comentário!

@efraimfilhopb @julianleamosdeputadofederalpb @depaguinaldoribeiro @dr.damiao @ednahenriqueoficial @freianastaciopt @gersvionamaia @hugomottapb @pedrocl @rui.carneiro @wellingtonrobertopb @wilsonsantiago

LABEL: no misinformation

Coronavírus: em 3 pontos, o que o Brasil pode perder no cenário internacional ao omitir dados de Covid-19 | Coronavírus | G1 <https://g1.globo.com/bemestar/coronavirus/noticia/2020/06/09/coronavirus-em-3-pontos-o-que-o-brasil-pode-perder-no-cenario-internacional-ao-omitir-dados-de-covid-19.ghtml>

LABEL: no misinformation

As Forças Armadas seguem na missão de enfrentamento ao coronavírus e de apoio à população brasileira nesse combate. Marinha, Exército e Aeronáutica estão unidas em 10 Comandos Conjuntos espalhados pelo Brasil. Até o momento, foram empregadas 60 embarcações, 800 viaturas, 26 aeronaves, além de 25 mil militares. Uma das necessidades imediatas é dar assistência a pessoas de baixa renda, pois, em decorrência do isolamento social, não podem realizar suas atividades de trabalho... \*Saiba mais através do link abaixo:\*

<https://dunapress.org/2020/04/07/militares-apoiam-campanhas-de-doacao-de-alimentos/>

LABEL: no misinformation

<http://rumasemnoticias.blogspot.com/2020/04/ilheus-prefeitura-de-ilheus-decide.html>

Prefeitura de Ilhéus decide manter comércio fechado.

Subiu pra 14 os casos confirmados em Ilhéus pela covid19.

Fonte: o autor.

"saúde", "medida" sugerem um texto de caráter informativo-educativo, em relação às medidas de prevenção da disseminação do vírus. Por outro lado, a forte presença dos termos "china", "globo", "governadores" sugerem a descrição de teorias conspiratórias e *fake news* que foram notavelmente associadas com esses termos e no contexto da pandemia.



## 5 O SISTEMA DE RECOMENDAÇÃO PROPOSTO

Neste capítulo será demonstrado o sistema de recomendação que foi proposto para o desenvolvimento deste trabalho, delineando sua arquitetura e mecanismos. Iremos expor detalhes de sua implementação bem como explicar as razões pela abordagem de sistema recomendativo escolhida, de acordo com o cenário proposto pelos dados.

### 5.1 Contexto e Iniciativa

Como meta deste trabalho, tem-se como objetivo o desenvolvimento de um sistema de recomendação que possa fazer associações entre conteúdos semelhantes, isto é, que os itens recomendados sejam *similares* ao item de entrada. No contexto de mensagens coletadas do WhatsApp, isto significaria que pudéssemos desenvolver um mecanismo que possa receber um texto de entrada e receber, do nosso conjunto de dados, aquelas mensagens mais parecidas à que foi fornecida. Ou seja, se uma mensagem com conteúdo *não-desinformativo* é fornecida como entrada, espera-se que uma lista de mensagens com conteúdo *semelhante* e com caráter *não-desinformativo* seja recomendada. O mesmo vale para as mensagens desinformativas: com uma mensagem desinformativa de entrada, espera-se receber mensagens desinformativas semelhantes.

Apesar da segunda expectativa (de receber mensagens desinformativas similares) não ser interessante para uma aplicação real que busca combater informação diretamente, ela ainda será importante para avaliar a capacidade em si do SR de encontrar mensagens semelhantes que utilizam um certo *critério de relevância pré-estabelecido* (que no caso é o de desinformação). Dessa forma, esse critério poderia ser ajustado de acordo com a necessidade e objetivo dessa eventual aplicação que atua no combate à desinformação. Essa questão é elencada mais adiante na Seção 7.1.

Como estamos lidando com *conteúdo* da mensagem, o tipo de sistema recomendador que mais faz sentido para a aplicação são os **sistemas de recomendação baseados em conteúdo**. Será tomada a iniciativa de considerar apenas o conteúdo das mensagens como mecanismo de cálculo de similaridade, sem levar em conta os outros atributos que a caracterizam. Como o conteúdo da mensagem é o único atributo que a descreve, a medição de similaridade entre as mensagens será a medição de similaridade do conteúdo delas. Em outras palavras, o problema de similaridade de *itens* num sentido genérico se reduz a um problema de similaridade de *textos*.

Para tornar as mensagens aplicáveis ao algoritmo será preciso realizar a transforma-

ção vetorial conforme explicada na Seção 2.2 e, neste sentido, iremos aplicar a técnica **TF-IDF**, que valoriza a importância das palavras e expressões em termos de frequência, um passo importante em direção à identificação do sentido das declarações. Para o cálculo de similaridade, será adotada a métrica de **similaridade do cosseno**, que é uma métrica que *independe da magnitude*, pois é basicamente uma métrica de comparação vetorial em um espaço normalizado. Ela também é relativamente simples de calcular, especialmente em conjunção com escores TF-IDF. Por fim, será construída uma função que é capaz de receber esses escores e utilizá-los para fazer as recomendações de mensagens similares.

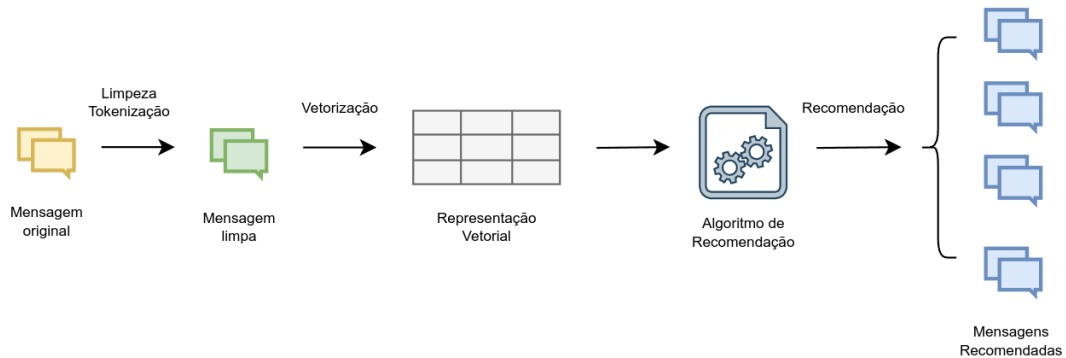
## 5.2 Arquitetura do SR

A disposição dos componentes do sistema recomendador faz uso de algumas das técnicas e de processamentos de dados já discutidos no Capítulo 4. Definimos basicamente um processo simples para esse sistema: recebemos a mensagem, aplicamos o processo de limpeza e *tokenização*, transformamos a mensagem na sua representação vetorial e calculamos a similaridade através de uma certa métrica, para só então processar as recomendações. Em detalhes, a seguinte lógica é executada:

1. **Pré-processamento:** Uma mensagem é fornecida como entrada. Essa mensagem passa por alguns dos processos de limpeza antes descritos no Capítulo 4 - remoção de links, remoção de espaços em branco e os caracteres são forçados para letras minúsculas;
2. **Vetorização:** Palavras vazias são desconsideradas do conteúdo do texto. A mensagem de entrada é integrada ao conjunto de dados original. O algoritmo vetorizador TF-IDF é utilizado para gerar a representação matricial de todas as mensagens.
3. **Cálculo da Similaridade:** Com base na matriz TF-IDF, aplicamos a métrica de *similaridade do cosseno* para gerar as métricas de semelhança das mensagens entre si;
4. **Recomendação:** O algoritmo de recomendação recebe a matriz das similaridades e, de acordo com a mensagem de entrada, ordena as restantes de acordo com seu escore de similaridade em relação à entrada. Ele retorna as top  $n$  mensagens mais similares.

A Figura 22 mostra uma visualização de alto nível deste processo.

Figura 22 – Visão de Alto Nível do Sistema de Recomendação



Fonte: o autor.

### 5.3 Implementação

A implementação de código do sistema de recomendação foi pensada para ser o mais simples possível, pelo fato de a complexidade da arquitetura escolhida ser baixa. Todo o código é escrito na linguagem Python, conforme mencionado no Capítulo 4, utilizando as bibliotecas citadas na Seção 4.3, e ele poderá ser encontrado em seu próprio repositório<sup>1</sup>, dentro do *notebook* `recommendation-algorithm-clean.ipynb`. Também foram utilizadas algumas bibliotecas nativas da linguagem, como o módulo `re`, que lida com expressões regulares. A primeira etapa do código incorpora estas bibliotecas auxiliares para que suas funções estejam disponíveis na definição dos algoritmos do SR em si.

#### 5.3.1 Pré-processamento

O primeiro passo é responsável por tornar a mensagem livre de fatores que possam adicionar complexidade desnecessária à modelagem. Por meio da aplicação do uso das expressões regulares e de algumas funções nativas da linguagem Python, aplicamos o seguinte processo:

1. **Removemos os *links***: Eles podem gerar entropia desnecessária por conterem expressões grandes e únicas que não adicionem significado real para a mensagem;
2. **Removemos os *espaços em branco***: retiramos ambiguidade das declarações;
3. **Forçamos as letras para o *formato minúsculo***: assim representamos de uma forma padronizada palavras que possuem o mesmo significado, mas que poderiam estar definidas em caixas de formatação diferentes (ex: *Ideia* e *ideia*).

Em seguida, foi criada uma função que aplica os três passos anteriores para aplicar a limpeza em um certo texto, ela é definida pelo algoritmo em alto nível abaixo.

<sup>1</sup> [https://gitlab.com/talesaraujo/msg-ranking-research/-/tree/revfinal?ref\\_type=heads](https://gitlab.com/talesaraujo/msg-ranking-research/-/tree/revfinal?ref_type=heads)

---

**Algoritmo 1:** Preprocessamento da mensagem
 

---

**Input** : *mensagem*
**Output** : *mensagem\_processada*
*mensagem\_processada*  $\leftarrow$  *mensagem*;

*mensagem\_processada*  $\leftarrow$  *remover\_links(mensagem\_processada)*;

*mensagem\_processada*  $\leftarrow$  *remover\_espacos\_em\_branco(mensagem\_processada)*;

*mensagem\_processada*  $\leftarrow$  *forçar\_minusculas(mensagem\_processada)*;

**return** *mensagem\_processada*;

---

### 5.3.2 Vetorização

Para aplicar esta etapa, assumimos que o corpus de texto tenha passado pelo processo de limpeza anterior, e tenha sido carregado em uma variável do tipo `DataFrame` da biblioteca `pandas`. No presente trabalho, o conjunto de dados representa o corpus pelo atributo `text`, portanto criamos um novo atributo chamado `clean_text` que representa o mesmo anterior porém após passar pelo processo de limpeza. Incorporamos a classe responsável pela abstração que realiza a vetorização; essa classe é implementada pela biblioteca `scikit-learn`.

Definimos a instância desta classe, que permite que possamos fornecer uma lista de palavras vazias para que elas sejam desconsideradas no algoritmo. Essa lista, representada aqui pela variável `STOPWORDS`, é obtida diretamente da biblioteca `NLTK`, que possui algumas dessas palavras definidas para a língua portuguesa. Uma vez que essa lista era relativamente incompleta, **adicionamos manualmente** algumas palavras vazias que aparentavam influenciar *negativamente* os dados na etapa de *análise exploratória*. Adicionamos a mensagem alvo ao corpus e aplicamos o processo de vetorização no novo corpus. Essa função retorna três itens: uma matriz esparsa com a representação vetorial das mensagens, uma cópia dos dados com o novo item inserido e uma referência para a nova mensagem dentro desse novo conjunto de dados. O Algoritmo 2 descreve, em bastante alto nível, como esses passos são executados.

### 5.3.3 Cálculo da Similaridade

Esse passo é efetuado de maneira relativamente simples: a biblioteca `scikit-learn` possui a função `cosine_similarity()` que implementa a métrica de similaridade do cosseno. Entretanto, como estamos aplicando o algoritmo TF-IDF, percebemos que o algoritmo poderá ficar mais rápido se aplicarmos a função `linear_kernel()` em vez da anterior, que calcula o produto interno de dois vetores. Essa função combinada com o TF-IDF irá resultar na

---

**Algoritmo 2:** Vetorização

---

**Precondition:** A mensagem de entrada precisa estar preprocessada.  
**Input :** conj\_dados, mensagem  
**Output :** matriz\_tfidf, novo\_conj\_dados, novo\_indice

```

if mensagem não for NIL then
  | novo_conj_dados ← conj_dados.adicionar_elemento(mensagem);
  | novo_indice ← novo_conj_dados.indice_ultima_posicao();
else
  | novo_conj_dados ← conj_dados;
  | novo_indice ← -1;
end
modelo_tfidf ← nova instancia do vetorizador;
matriz_esparsa_vetorial ←
  modelo_tfidf.executar_transformacao(novo_conj_dados['clean_text']);
return matriz_esparsa_vetorial, novo_conj_dados, novo_indice;

```

---

similaridade do cosseno, pois o algoritmo do TF-IDF já aplica uma normalização-L2 (a mesma do denominador da similaridade do cosseno). A função responsável por essa etapa retornará a matriz  $n \times n$  que representa os escores de similaridade, onde  $n$  é o número de mensagens do corpus. O Algoritmo 3 apresenta a definição de alto nível desta etapa, que utiliza uma das variáveis de retorno do Algoritmo 2, a matriz que possui a representação vetorial das palavras do conjunto de dados.

---

**Algoritmo 3:** Cálculo de Similaridade

---

**Input :** matriz\_esparsa\_vetorial  
**Output :** matriz\_de\_similaridade

```

matriz_de_similaridade ←
  calculo_kernel_linear(matriz_esparsa_vetorial, matriz_esparsa_vetorial);
return matriz_de_similaridade;

```

---

### 5.3.4 Função de Recomendação

Essa última etapa é basicamente uma definição de algumas funções auxiliares que recebem a matriz com os escores de similaridade e faz algumas manipulações com os seus índices de forma a trazer as  $k$  mensagens mais semelhantes. Por meio dos seus índices no conjunto de dados, elas mostram no final quais são essas mensagens. O Algoritmo 4 demonstra a parte mais relevante da função de recomendação, que indexa apenas os valores selecionados pelo índice da mensagem do conjunto de dados. Nele,  $k$  representa o número de recomendações.

---

**Algoritmo 4:** Função de Recomendação

---

**Input** : conjunto\_de\_dados, indice, matriz\_de\_similaridade, numero\_recomendacoes**Output** : escores\_de\_similaridade**if** *indice* for válido no conjunto\_de\_dados **then**    *escores\_de\_similaridade*  $\leftarrow$  *matriz\_de\_similaridade*[*indice*];    *escores\_de\_similaridade*  $\leftarrow$  *vetor\_indexado*(*escores\_de\_similaridade*);    *escores\_de\_similaridade*  $\leftarrow$  *ordem\_indice\_reversa*(*escores\_de\_similaridade*);    *escores\_de\_similaridade*  $\leftarrow$  *subvetor*(*escores\_de\_similaridade*)[2 :  
        (*numero\_recomendacoes* + 1)];    **return** *escores\_de\_similaridade*;**return** *NIL*;

---

O próximo capítulo irá fazer as demonstrações de funcionamento do sistema de recomendação proposto, bem como avaliá-lo de acordo com os critérios estabelecidos pelas métricas de avaliação de desempenho mencionadas no Capítulo 2.

## 6 AVALIAÇÃO EXPERIMENTAL

A partir de agora, serão feitas demonstrações de uso do sistema recomendador definido pela arquitetura do Capítulo 6. Pretendemos, com esta seção, vislumbrar o entendimento de duas questões:

1. O quão *bem* o sistema de recomendação atua para recomendar as mensagens?
2. Seria possível utilizar o sistema de recomendação como uma espécie de classificador para mensagens sem rótulo?

Para responder à primeira pergunta, iremos demonstrar o SR em ação com algumas mensagens e também pretendemos recorrer às métricas de avaliação para Sistemas de Recomendação descritas no Capítulo 2 para avaliar a sua performance, de forma a visualizar como o critério de utilidade com base em conteúdo desinformativo penaliza o escore definido por cada métrica.

Para o caso da segunda pergunta, pretende-se argumentar com base nas demonstrações visuais da etapa anterior para que, com o caráter de um classificador, se possa avaliar a possibilidade de o sistema de recomendação ser capaz de rotular mensagens ainda não vistas, isto é, mensagens que poderiam representar conteúdo totalmente novo ainda não contemplado pelo sistema, visto que elas não estariam sendo utilizadas como parte do treinamento nem possuem rótulo de desinformação atribuído. A avaliação deste classificador será feita com as métricas tradicionais utilizadas em problemas de classificação de aprendizado de máquina.

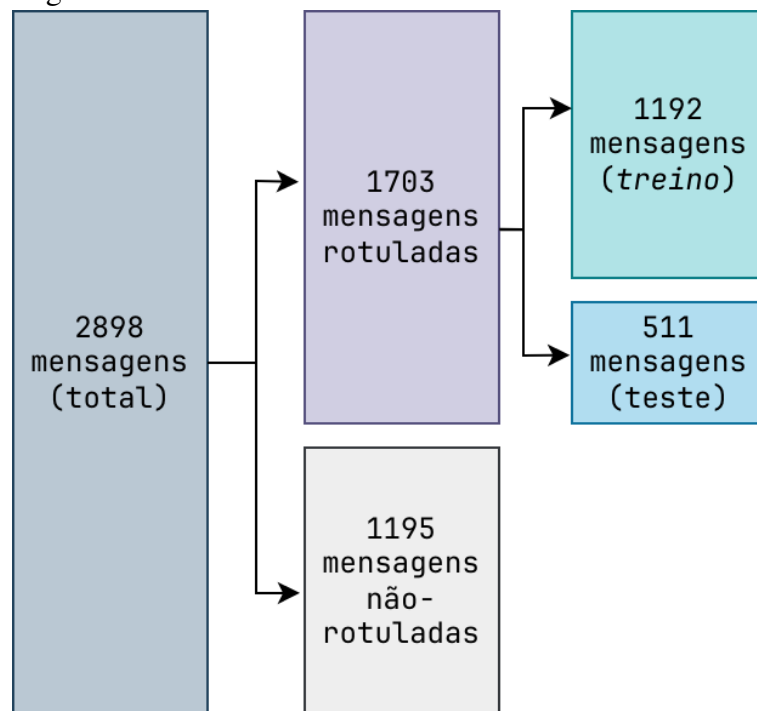
Por fim, pretendemos delinear limitações do sistema e pontos onde ele poderá ser desenvolvido tendo em vista a superação dessas limitações.

### 6.1 Experimentos

O conjunto de dados no total contém 2898 mensagens, compreendendo mensagens com e sem rótulo (atributo que define de maneira explícita a presença de desinformação). Para realizar o cálculo da matriz de recomendações, fizemos uma separação do conjunto de dados de acordo com a presença do rótulo e, com isso, apenas as mensagens **rotuladas** foram utilizadas para este efeito, contabilizando um total de 1703 mensagens. Partindo do subconjunto de mensagens rotuladas, criamos aleatoriamente mais dois subconjuntos deste, com a proporção 70%-30%: um de *treino* e *teste*, ambos contendo 1192 e 511 mensagens, respectivamente. A Figura 23 ilustra o processo de divisão dos dados.

A princípio, apenas as mensagens do conjunto de treino (em *verde*, na Figura 23) foram utilizadas para o cálculo da matriz de recomendações. Entretanto, o algoritmo de recomendação foi designado de forma que pudéssemos listar recomendações tanto para mensagens deste conjunto como para mensagens que não pertencem a ele. As mensagens deste conjunto são fornecidas através de seus índices, para questões de identificação e, são, obviamente, rotuladas. Decidimos utilizar as mensagens não-rotuladas (em *cinza-claro*, na Figura 23) como mensagens completamente alheias ao corpus de treinamento, para considerar um cenário de uma mensagem real, isto é, uma mensagem totalmente nova e ainda não contemplada pela etapa de avaliação *offline* do sistema de recomendação. Serão realizadas duas demonstrações diferentes: a primeira utilizará uma mensagem do corpus de treinamento (isto é, rotulada) e a segunda uma mensagem não rotulada.

Figura 23 – Divisão dos dados



Fonte: o autor.

## 6.2 Running Examples

As Figuras 24 e 25 demonstram a escolha de uma mensagem aleatória do corpus de treinamento (conjunto *verde*). Escolhemos o índice aleatório  $e$ , de acordo com ele, ao fornecê-lo como argumento de entrada ao algoritmo, ele irá retornar as  $k$  mensagens mais similares. Para o simples propósito de demonstração, podemos escolher um valor qualquer para  $k$ . Usaremos

$k = 3$ .

### Figura 24 – Mensagem aleatória selecionada do corpus de treino

Mensagem de Índice 64 | Classe 1.0

URGENTE: Descoberto o “gabinete do ódio” do governo chinês, contas falsas criadas para falar bem da China no Twitter e influenciar a opinião pública mundial.

\*Comunistas são baixos\*.

<https://portalnovoporte.com.br/?p=26673&amp=1>

Fonte: o autor.

### Figura 25 – Lista de recomendações para a mensagem da Figura 24

LISTA DE RECOMENDAÇÕES

Rank 1 | Índice 301 | Similaridade 0.108 | Classe: 1.0

Os nossos políticos precisam agir assim para proteger o nosso povo dos da globolixo e dos comunistas que estão entregando o nosso Brasil para os comunistas chineses.

Rank 2 | Índice 70 | Similaridade 0.107 | Classe: 1.0

URGENTE SR PRESIDENTE SALVE O BRASIL E SEU POVO, ESTAMOS, MORRENDO POR VIRUS CHINÊS A DECAD

Rank 3 | Índice 39 | Similaridade 0.106 | Classe: 1.0

Organização Mundial de Saúde: O aborto é “essencial” durante a pandemia de coronavírus chinês.

<https://conexaopolitica.com.br/ultimas/organizacao-mundial-de-saude-o-aborto-e-essencial-durante-a-pandemia-de-coronavirus-chines/amp/>

Fonte: o autor.

Percebemos que a mensagem aleatória em escolha possui classe atribuída como conteúdo desinformativo (rótulo 1). Podemos sinalizar, na Figura 25, a presença de certos termos mais expressivos e alarmistas, como “*comunistas*”, “*governo chinês*”. Podemos identificar termos semelhantes na lista de recomendações, o que sugere que as mensagens estão, até certo ponto, parecidas no sentido lexical, pela presença das variações do termo “*chinês*”.

A Figuras 26 e 27 fazem outra demonstração com outra mensagem aleatória escolhida do mesmo corpus de treinamento. Essa mensagem possui rótulo diferente da anterior e seu conteúdo cita informações sobre o “*auxílio-emergencial*”, medida de apoio financeiro empregada pelo Governo Brasileiro durante o período pandêmico.

Mais uma vez, as recomendações aparentam ter semelhança lexical (como demonstrado pela Figura 27), pela simetria dos termos compartilhados entre as mensagens, que mencionam o mesmo tópico. Neste caso, o sentido semântico também é o mesmo: informar sobre ações envolvendo o “*auxílio-emergencial*”. Portanto, para essa amostra, o cálculo de similaridade léxico foi suficiente para sustentar uma recomendação semântica similar.

## Figura 26 – Mensagem aleatória selecionada do corpus de treino

Mensagem de Índice 1159 | Classe 0.0

Olá bom dia!  
Este é o link do auxílio emergencial para autônomos , micro empresário e desempregados  
Esse site é o site/link oficial da caixa para realizar o cadastro para recebimento do auxílio emergencial COVID-19 de R\$600 reais.

Secretaria Municipal de assistência social

<https://auxilio.caixa.gov.br/#/inicio>

Fonte: o autor.

## Figura 27 – Lista de recomendações para a mensagem da Figura 26

LISTA DE RECOMENDAÇÕES

Rank 1 | Índice 1268 | Similaridade 0.224 | Classe: 0.0

\*Auxílio emergencial: aplicativo para cadastro estará disponível na terça-feira (7) Tire suas dúvidas sobre o 'corona voucher'\*

<http://surgiu.com.br/2020/04/06/auxilio-emergencial-aplicativo-para-cadastro-estara-disponivel-na-terca-feira-7/>

Rank 2 | Índice 44 | Similaridade 0.213 | Classe: 0.0

Trabalhador deve receber auxílio emergencial em até 48 horas A partir das 9h de hoje (7), de 15 milhões a 20 milhões de trabalhadores informais não inscritos em programas sociais poderão baixar o aplicativo da Caixa Econômica Federal que permitirá o cadastramento para receberem a renda básica emergencial. O auxílio - de R\$ 600 ou de R\$ 1,2 mil para mães solteiras - será pago por pelo menos três meses para compensar a perda de renda decorrente da pandemia de corona vírus... \*Saiba mais através do link abaixo:\*

<https://dunapress.org/2020/04/07/autonomo-pode-baixar-aplicativo-a-partir-de-hoje-para-renda-de-r-600/>

Rank 3 | Índice 49 | Similaridade 0.194 | Classe: 0.0

Ministro detalhou ações prioritárias da pasta e enfatizou o papel do Auxílio Emergencial, do Bolsa Família e do SUAS. O Nordeste como destaque nos números superlativos do Bolsa Família. A abrangência do Auxílio Emergencial em todo o território nacional, com mais de 51 milhões de brasileiros que já receberam duas das três parcelas previstas em lei. A adesão de 4.737 municípios a equipamentos e recursos repassados ao Sistema Único de Assistência Social (SUAS)...

<https://dunapress.org/2020/06/09/durante-reuniao-do-conselho-do-governo-ministro-da-cidadania-destaca-importancia-das-acoes-no-combate-a-covid-19/>

Fonte: o autor.

A seguir, na Figura 28, demonstramos mais uma vez o algoritmo fornecendo como entrada uma mensagem (não-rotulada, do conjunto *cinza-claro* da Figura 23) que inicialmente não fazia parte do conjunto inicial utilizado anteriormente. Ela é integrada a este conjunto e suas recomendações são enumeradas de acordo com a Figura 29.

## Figura 28 – Mensagem aleatória sem rótulo (mensagem real)

Todo meu apoio aos \*estudantes de medicina\* brasileiros que podem nos ajudar a vencer o \*COVID-19\* e o caos na \*saúde pública\*. São 15000 novos médicos no mercado de trabalho, jovens, \*corajosos\* e \*patriotas\* que querem ajudar ao povo brasileiro nesse momento tão difícil. \*Todo meu apoio e vamos à luta!\* 🇺🇵 🇧🇷

<https://www.instagram.com/p/B-VduADjQRb/?igshid=xpzkttoy5tg0u>

\*Faça parte da nossa lista de Transmissão adicionando esse contato e solicitando receber informações do mandato no privado!\*

Fonte: o autor.

Por uma inspeção visual e interpretação vemos que a mensagem original refere-se a um suposto apoio a estudantes da carreira de medicina que buscavam acelerar seu processo

Figura 29 – Lista de recomendações para a mensagem da Figura 28

LISTA DE RECOMENDAÇÕES			
Rank 1	Índice 2	Similaridade 0.185	Classe: 0.0
<p>É com muita alegria que comunico que mais um pedido nosso relacionado ao Coronavírus foi atendido! O Ministério da Educação decidiu adiantar a formatura dos estudantes de Medicina, Enfermagem e outros cursos da área da Saúde para atuarem nos hospitais no enfrentamento da Covid-19! São mais de 10 mil universitários jovens e corajosos que com certeza querem se juntar às equipes médicas que atuam na linha de frente do combate ao Coronavírus!</p> <p>Obrigado, MEC! E eu continuo em contato com reitores de universidades e faculdades para conseguir mais apoio para essa importante iniciativa!</p> <p><a href="https://www.instagram.com/p/B-pnWueIqL2/?igshid=1je4qpn5hczy">https://www.instagram.com/p/B-pnWueIqL2/?igshid=1je4qpn5hczy</a></p>			
Rank 2	Índice 681	Similaridade 0.131	Classe: 0.0
<p>Chegada dos médicos e enfermeiros pra ajudar no combate ao coronavírus no df. 🇧🇷</p>			
Rank 3	Índice 1429	Similaridade 0.124	Classe: 0.0
<p>Como a atual Pandemia do Covid-19 está gerando muita ansiedade e medo pelo risco de contaminação, isolamento, desemprego e muita incerteza sobre o futuro. Esse e-book incrível vai te ajudar no controle das emoções nesse momento difícil, repleto de atividades e exercícios para que você possa colocar em prática e assim desenvolva equilíbrio emocional em meio ao caos. Vamos juntos?!</p> <p>E-book disponível em <a href="https://www.najmapsico.com.br/covid-19/">https://www.najmapsico.com.br/covid-19/</a></p> <p>Divulguem! Todos devem ficar em casa e podem ficar bem, com saúde emocional!</p>			

Fonte: o autor.

de formação em visto ao contexto da escassez de profissionais qualificados da área por razão das demandas da pandemia. As duas primeiras recomendações claramente falam a respeito do mesmo assunto, enquanto que a terceira não está diretamente ligada a este tópico. Percebemos portanto que, apesar de as três identificarem vocabulários semelhantes, a semântica é similar em apenas duas delas. Notamos também que todas as mensagens sugeridas aparentam *não constar conteúdo desinformativo*, o que pode apontar a leve *possibilidade* da mensagem de entrada, apesar de não-rotulada, ser da mesma classe. Poderíamos utilizar esta observação como hipótese para basear a construção do classificador de mensagens não-rotuladas, o que será avaliado mais adiante.

### 6.3 Resultados

Até o momento percebemos que, pelo menos lexicamente, o algoritmo recomendador foi capaz de retornar mensagens similares umas às outras de acordo com sua semelhança no vocabulário. Algumas delas, apesar de ainda apresentarem semelhanças do palavreado, não aparentam discutir sobre o mesmo assunto. Ainda assim, possuímos grande chance de, apenas com esse critério de semelhança, recuperar mensagens parecidas que falem do mesmo assunto.

Consideramos que, para este trabalho, um cenário ideal seria um sistema recomen-

dador que pudesse retornar mensagens semelhantes, que falem do mesmo assunto e sejam da mesma classe. Ou seja, se uma mensagem desinformativa que fala sobre métodos de tratamento não-comprovados fosse fornecida como entrada, esperamos que uma lista de mensagens similares, também desinformativas, fosse elencada na lista de recomendações, e que ainda falasse a respeito do mesmo assunto. Da mesma forma, se fornecêssemos uma mensagem informativa acerca de métodos de proteção contra a doença, esperamos que mensagens a respeito de métodos de prevenção sejam retornadas. Mas como avaliar esse critério subjetivo de *utilidade*?

Bem, para nós, uma mensagem útil seria uma que fosse retornada com as características acima. Poderíamos definir um primeiro critério de utilidade para fazer uso das recomendações, e esse critério é o de **desinformação**: com base nas medidas de avaliação de desempenho descritas no Capítulo 2, vamos listar alguns resultados de acordo com essas métricas, que avaliam, em um certo nível, essa utilidade.

### 6.3.1 Avaliação da Performance do SR

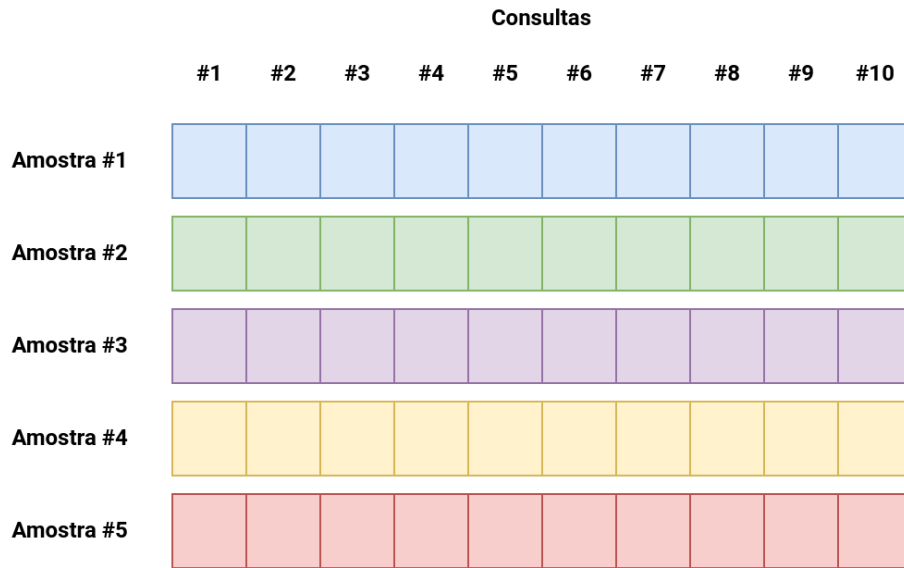
Para executar a avaliação do desempenho do Sistema de Recomendação de acordo com o critério de utilidade definido acima, considerando que queremos as mensagens mais similares no **topo da lista**, nos voltamos às métricas de avaliação offline como primeiro passo, mais especificamente àquelas que são bastante sensíveis ao ordenamento da lista de recomendações: são elas o **Mean Reciprocal Rank**, ou MRR, e a **Mean Average Precision**, ou MAP. Desta forma, referenciaremos a primeira questão sugerida no início deste capítulo.

Como essas métricas são calculadas em cima de *consultas*, isto é, são calculadas em várias rodadas de pedidos de consulta de recomendação, selecionamos aleatoriamente cinco listas de mensagens diferentes do conjunto de treino, cada lista com dez índices. Cada índice representa uma mensagem no conjunto de dados, dessa forma, uma *consulta* diferente. A Figura 30 ilustra o processo de amostragem. Nela, cada quadrado representa um índice diferente, isto é, uma nova consulta.

Definimos o valor de  $K$ , o número de recomendações, como os valores fixos 1, 3 e 5. Para *cada um* desses valores e para *cada uma* das cinco listas, calculamos os valores de MRR e MAP.

Com esses valores, temos uma visão de como o sistema recomendador se comportou para as cinco listas cada qual com dez mensagens, em duas perspectivas diferentes: com os valores de MRR, temos a ciência do quão bem o sistema colocou o tipo de mensagem esperada

Figura 30 – Amostragem de Consultas ao SR



Fonte: o autor.

no topo da lista. Em outras palavras, indica a qualidade do sistema em recomendar a *melhor* mensagem primeiro antes das outras. Por outro lado, a MAP indica o quão equilibradas estão as indicações de acordo com suas ordens de prioridade de recomendação. Por exemplo, uma lista de recomendação que aponta um item não-relevante na posição 1 é penalizada muito mais fortemente do que uma que possui um item não-relevante na posição 3, assim como um item de outra lista de recomendação que está bem colocado na posição 1 afetará mais positivamente o valor do escore da sua lista do que o de outra lista de recomendação onde uma mensagem relevante foi encontrada apenas na posição 3. A Tabela 6 demonstra os resultados obtidos com a execução destes cálculos.

Tabela 6 – Avaliação da performance

	K=1		K=3		K=5	
	<b>MRR</b>	<b>MAP</b>	<b>MRR</b>	<b>MAP</b>	<b>MRR</b>	<b>MAP</b>
1	0.60	0.60	0.73	0.73	0.75	0.76
2	0.70	0.70	0.85	0.87	0.85	0.87
3	0.60	0.60	0.75	0.75	0.75	0.74
4	0.90	0.90	0.95	0.94	0.95	0.92
5	0.50	0.50	0.65	0.66	0.68	0.70
<b>Média</b>	<b>0.66</b>	<b>0.66</b>	<b>0.79</b>	<b>0.79</b>	<b>0.80</b>	<b>0.80</b>

Fonte: elaborado pelo autor.

No geral, percebemos valores semelhantes em grande parte dos casos. Para os valores de  $K$  igual a 1, o valor das métricas foi, em média, relativamente menor se comparado aos outros dois casos. Poderíamos interpretar que a lista de recomendações com esse tamanho ainda não

possui quantidade de itens o suficiente para dar uma boa avaliação que possa corresponder aos critérios descritos acima: para  $K = 1$ , o critério de relevância é avaliado binariamente, portanto não é tão bom para avaliar a presença do "melhor recomendado", pro caso da MRR, e nem é bom pra avaliar os *rankings* em conjunto, como propõe a MAP. Ainda assim, na média, os valores indicam que o algoritmo recomendador indica bem em mais da metade das ocorrências, trabalhando melhor que um algoritmo que escolhe as listas aleatoriamente, pelo menos para esse valor de  $K$ . Por outro lado, os valores de MRR e MAP tendem a possuir valores praticamente idênticos para  $K = 1$ , e isto é concebível, considerando que nesta condição, não há distinção das características das métricas: a relevância é avaliada binariamente para a MRR, que se transforma em uma simples média harmônica para cada um dos itens da consulta, e a ideia da média harmônica também se reflete para a MAP, já que seu cálculo de precisão considera apenas o item relevante por rodada, que será obrigatoriamente 1, se for relevante, claro.

À medida que o valor de  $K$  aumenta, os valores começam a se diferenciar e as características das métricas começam a ficar um pouco mais evidentes: vemos diferentes cenários para cada uma das amostras de lista de consulta, onde algumas apresentam boa performance, mas outras nem tanto. Por exemplo, na lista de consultas de recomendação número 4, se considerarmos o tamanho dessa lista de recomendação como 5 (isto é,  $K = 5$ ), vemos que sua habilidade de encontrar a *primeira melhor mensagem* foi melhor do que sua capacidade de ter um *conjunto equilibrado* de recomendações, como mostram os escores MRR=0,95 e MAP=0,92. Para outros resultados, essa dinâmica se inverte, e isto pode se dever a diversos fatores peculiares dessas listas de consultas. Na média, temos um bom resultado crescente em  $k$  que evidencia que nosso sistema recomendador foi capaz de retribuir relevância o suficiente se considerarmos um leque maior de recomendações feitas para cada lista.

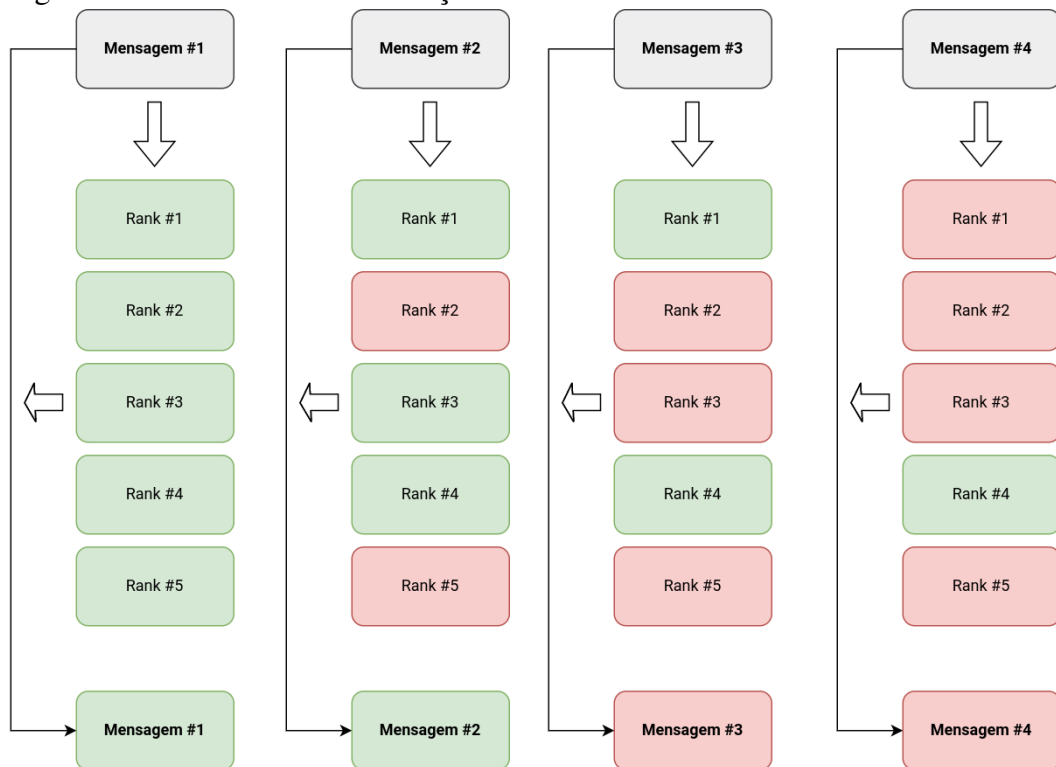
### 6.3.2 Avaliação do SR como Sistema Rotulador

Visando responder à segunda pergunta proposta no início deste capítulo, iremos doravante avaliar a capacidade de um sistema baseado no SR proposto atuar como classificador de mensagens supostamente não-rotuladas, isto é, com base no conjunto de teste (em *azul*, na Figura 23).

Para isto, podemos fixamos o valor do tamanho da lista de recomendações para o melhor desempenho médio de MRR e MAP, isto é,  $k = 5$ . Desta forma, como critério de classificação, podemos aplicar a contagem de cada rótulo para decidir se a mensagem de

entrada, não-rotulada, é desinformação ou não. Por exemplo, se a cada vez que fornecermos uma mensagem de entrada e obtermos sua lista de recomendações, como descrito na Figura 31, poderíamos rotular a mensagem de entrada como desinformação se a quantidade de itens recomendados como desinformação for maior ou igual a 3. A Figura 31 indica como *cinza* mensagens não-rotuladas, *vermelho* como mensagens que contém desinformação e *verde* para as que não contém.

Figura 31 – Critério de Classificação



Fonte: o autor.

Estabelecido o critério, executamos, para cada mensagem presente no conjunto de teste (em *azul*, na Figura 23), o processo de recomendação de mensagens para a mensagem “não-rotulada” em questão. Da lista de recomendações utilizamos o critério anterior para classificar a mensagem. Aqui tomamos o cuidado de separar previamente o conjunto de teste como uma porção das mensagens rotuladas para simular as não-rotuladas, de forma que pudéssemos testar sua capacidade de acordo com os métodos de avaliação tradicionais de desempenho para problemas de classificação: **acurácia**, **precisão**, **revocação** e **score-F1**. Também computamos a exibição da *matriz de confusão* do classificador. Ao executarmos o processo descrito acima, obtemos os seguintes valores:

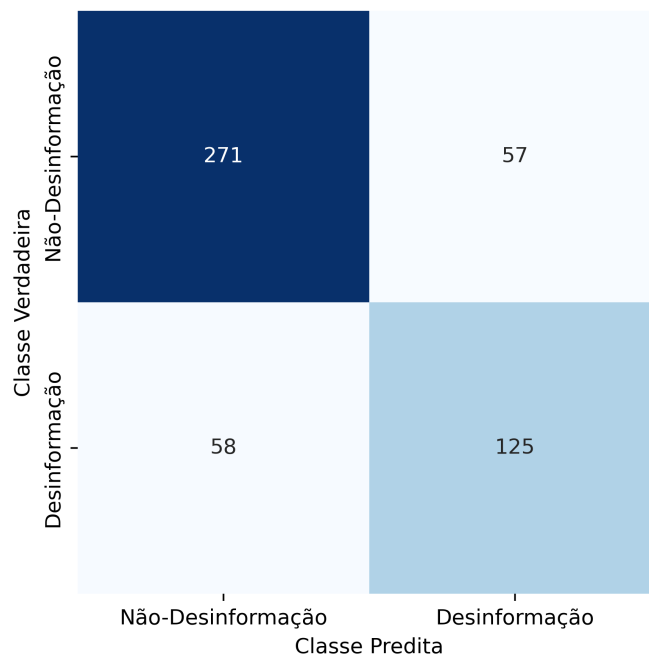
A métrica de *acurácia* indica de um modo geral como o classificador está se com-

Tabela 7 – Desempenho do classificador

Métrica	Valor
Acurácia	0.77
Precisão	0.69
Revocação	0.68
Escore-F1	0.68

Fonte: elaborada pelo autor.

Figura 32 – Matriz de Confusão



Fonte: o autor.

portando: 77% das mensagens do conjunto de teste foram classificadas corretamente. Entretanto, apesar de este indicador apresentar um valor relativamente alto, sabemos que 23% das mensagens foram rotuladas incorretamente. Isto se reflete nos valores menores para os indicativos de *precisão* e *revocação*, que são *praticamente* equivalentes - o *escore-F1* com valor similar aos dois anteriores sumariza isto. Isto significa que uma quantidade considerável de mensagens, originalmente de conteúdo não-desinformativo, foi classificada como contendo desinformação, e vice-versa.

Outro ponto que vale a pena ressaltar é que, se observarmos a matriz de confusão gerada pelo teste de performance do modelo, o classificador foi mais eficaz em classificar mensagens como *não-desinformativas* do que como *desinformativas*. Uma grande possibilidade é que isto possa se dever ao considerável *desbalanceamento das classes*, conforme apontado pela Figura 15, pois temos praticamente o dobro de exemplos de mensagens rotuladas como

desinformação. Entendemos que o modelo possuiu mais amostras para fazer o processo de classificação para a classe mais presente e isto se refletiu na sua execução.

No geral, o rotulador proposto conseguiu se sair bem na maior parte das situações, o que o torna *bem* melhor que um rotulador aleatório.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, discorremos sobre os mecanismos da desinformação, apresentamos um dos elementos básicos da área de Processamento de Linguagem Natural, que é o processo de vetorização, e apresentamos as definições teóricas de um Sistema de Recomendação, detalhando as suas principais vertentes de aplicação na literatura. Enumeramos as suas vantagens de cada aplicação, bem como seus empecilhos, e descrevemos como modelos híbridos de sistemas recomendativos tem ganhado espaço na implementação de arquiteturas complexas utilizados hoje em larga escala. Apresentamos as formas matemáticas de se computar semelhança entre itens de um sistema recomendativo, e descrevemos as diferentes maneiras de realizar procedimentos e processos avaliativos relacionados ao desempenho e a performance de sistemas desta natureza.

Realizamos um procedimento de análise exploratória sobre dados coletados a partir da plataforma WhatsApp voltados à pandemia de COVID-19, com um processo de limpeza, tratamento e análise de dados voltados à criação de atributos e mecanismos a partir dos já existentes, com o objetivo de descobrir tendências a respeito de certos aspectos encontrados nas mensagens que pudessem (ou não) caracterizar desinformação.

Com base nessas premissas teóricas, buscamos a construção de um sistema recomendador que fosse capaz de fazer distinções entre conteúdos desinformativos a partir de recomendações semelhantes no que concerne ao seu tipo de classe: ao fornecermos mensagens como entrada de caráter não-desinformativo, esperávamos obter mensagens semelhantes do mesmo caráter, e vice-versa. Nosso critério de similaridade entre as mensagens baseou-se em tão somente a semelhança textual entre os conteúdos dos textos das mensagens, não se valendo de mais nenhum atributo adicional para o cálculo da similaridade.

Ao realizarmos o procedimento de avaliação experimental, descobrimos primeiro, por inspeção visual, que o sistema recomendador proposto apresentou bom critério de similaridade para recomendar mensagens com características léxicas parecidas, isto é, com vocabulários semelhantes. Percebemos que a análise de similaridade com base nesse critério foi *importante* para avaliar o contexto sintático das expressões, porém não necessariamente foi o suficiente para fazer esse tipo de diferenciação. No geral, após executarmos os procedimentos de avaliação de desempenho do SR, com o cálculo de MRR e MAP, descobrimos que utilizar o critério de similaridade apenas baseado no léxico das palavras apresentava uma boa performance, mesmo para valores reduzidos de listas de recomendação. Portanto, tivemos a percepção de que a similaridade textual tem um papel *significativo* para avaliar as características que distinguem

textos desinformativos de textos não-desinformativos, e foi importante para construir um sistema que pudesse recomendar mensagens semelhantes que performa melhor que um gerador de mensagens aleatório. Portanto, na tentativa de responder às duas questões iniciais do Capítulo 6, ou seja,

- Q1: *O quão bem o sistema de recomendação atua para recomendar as mensagens?*

Considerando-se todos os critérios de processamento utilizados, o sistema de recomendação proposto **desempenhou bem** a sua tarefa de listar as mensagens similares, apresentando valores médios de MRR e MAP de 0.66, 0.79 e 0.80 para os tamanhos de lista de recomendação 1, 3 e 5, respectivamente. Valores de  $k$  maiores, i.e., com tamanho maior de lista de recomendação, trouxeram mais *utilidade* com suas listas de recomendações ao se tentar recuperar mensagens mais relevantes.

- Q2: *Seria possível utilizar o sistema de recomendação como uma espécie de classificador para mensagens sem rótulo?*

**Perfeitamente**, uma vez que o rotulador proposto baseado em listas de recomendação de cinco itens apresentou acurácia de 77%. Entretanto, não temos como concluir que esta é a *melhor* abordagem para a rotulação, uma vez que uma quantidade considerável de mensagens foi classificada erroneamente, como evidenciado pelo score-F1 relativamente menor do que o de acurácia, e isto poderia, de fato, gerar cenários indesejados. Percebemos também que a performance do classificador foi afetada pelo notável desbalanceamento das classes. Podemos concluir que este classificador pode servir de base para um eventual processo de rotulação mais elaborado.

Com isto em mente, temos a ciência de que sistemas de recomendação podem ser opções para serem trabalhadas e aplicados em contextos que trabalham desinformação com o objetivo de descobrir padrões e desenvolver soluções que possam criar um critério de discernimento entre comportamentos sociais onde há muita troca de informação envolvida, porém pouca moderação de qualidade de conteúdo.

## 7.1 Trabalhos Futuros

O sistema de recomendação desenvolvido pelo presente trabalho é uma aplicação dos modelos orientados a conteúdo. Entretanto, essa aplicação considerou apenas o conteúdo textual em si para a análise do critério de similaridade. Existem diversos outros atributos trabalhados no texto, bem como outros que podem ser derivados a partir de critérios fundamentados que podem

ser desenvolvidos para aumentar o número de características indicativas de similaridade entre os atributos, além da léxica e da frequência, como proposta por este trabalho. Por exemplo, o desenvolvimento de alguma análise que busca trabalhar o reconhecimento de entidades nomeadas dentro do texto e construir atributos em cima dessas identificações, poderia adicionar um peso ainda maior para a identificação de conteúdo desinformativo, como foram recorrentes as menções a entidades governamentais e alvos ataques de desinformação presentes nos conteúdos dos textos das mensagens.

Outra característica que vale a pena ressaltar é que, a princípio, foi pretendido, com o conjunto de dados utilizados, que um usuário pudesse entrar com uma notícia *falsa* no sistema de recomendação e pudesse receber recomendações de notícias *verdadeiras* relacionadas a ela (isto é, com base nos critérios de similaridade). Isto não foi colocado em prática por motivos de **não termos em mãos uma base de notícias verdadeiras**.

A ideia-chave de um sistema com tais características poderia ser transplantada em um sistema de recomendação de notícias, onde esse sistema poderia consumir uma API pública que fornece o título e o conteúdo, e o sistema teria a tarefa de identificar características desinformativas a partir de atributos desenvolvidos em cima dos dados da API, podendo assim, com base em uma notícia inicial, acessada por um usuário arbitrário, sugerir notícias semelhantes que não possuam características desinformativas dentro de seu conteúdo.

No caso de uso do classificador utilizado no processo de rotulação das mensagens, verificou-se o notável desbalanceamento das classes, o que inevitavelmente afetou a performance do mesmo. O uso de técnicas que buscam atenuar este efeito, como *undersampling*, *oversampling* ou SMOTE poderia se provar útil para corrigir este problema.

## REFERÊNCIAS

- ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. **IEEE Transactions on Knowledge and Data Engineering**, v. 17, n. 6, p. 734–749, 2005.
- AGGARWAL, C. C. *et al.* **Recommender systems**. [S. l.]: Springer, 2016. v. 1.
- BALE, J. M. Political paranoia v. political realism: On distinguishing between bogus conspiracy theories and genuine conspiratorial politics. **Patterns of prejudice**, Taylor & Francis, v. 41, n. 1, p. 45–60, 2007.
- BARAN, R.; DZIECH, A.; ZEJA, A. A capable multimedia content discovery platform based on visual content analysis and intelligent data enrichment. **Multimedia Tools and Applications**, v. 77, n. 11, p. 14077–14091, Jun 2018. ISSN 1573-7721. Disponível em: <https://doi.org/10.1007/s11042-017-5014-1>.
- BERINSKY, A. J. Rumors and health care reform: Experiments in political misinformation. **British Journal of Political Science**, Cambridge University Press, v. 47, n. 2, p. 241–262, 2017.
- BERKOVSKY, S.; KUFLIK, T.; RICCI, F. Mediation of user models for enhanced personalization in recommender systems. **User Modeling and User-Adapted Interaction**, Springer, v. 18, n. 3, p. 245–286, 2008.
- BERKOVSKY, S.; KUFLIK, T.; RICCI, F. Cross-representation mediation of user models. **User Modeling and User-Adapted Interaction**, Springer, v. 19, n. 1, p. 35–63, 2009.
- BILLSUS, D.; PAZZANI, M. Learning probabilistic user models. In: CITESEER. **UM97 Workshop on Machine Learning for User Modeling**. [S. l.], 1997.
- BREESE, J. S.; HECKERMAN, D.; KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. **arXiv preprint arXiv:1301.7363**, 2013.
- BRUSILOVSKI, P.; KOBASA, A.; NEJDL, W. **The adaptive web: methods and strategies of web personalization**. [S. l.]: Springer Science & Business Media, 2007. v. 4321.
- BURKE, R. Hybrid web recommender systems. In: \_\_\_\_\_. **The Adaptive Web: Methods and Strategies of Web Personalization**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 377–408. ISBN 978-3-540-72079-9. Disponível em: [https://doi.org/10.1007/978-3-540-72079-9\\_12](https://doi.org/10.1007/978-3-540-72079-9_12).
- BURKE, R.; FELFERNIG, A.; GÖKER, M. H. Recommender systems: An overview. **Ai Magazine**, v. 32, n. 3, p. 13–18, 2011.
- CABRAL, L. Fakewhatsapp. br: detecção de desinformação e desinformadores em grupos públicos do whatsapp em pt-br. 2021.
- CABRAL, L.; MONTEIRO, J. M.; SILVA, J. W. F. da; MATTOS, C. L. C.; MOURAO, P. J. C. Fakewhastapp. br: Nlp and machine learning techniques for misinformation detection in brazilian portuguese whatsapp messages. In: **ICEIS (1)**. [S. l.: s. n.], 2021. p. 63–74.
- CANAVILHAS, J.; BITTENCOURT, M.; ANDRADE, M. A. A. d. Conteúdos virais no facebook: estudo de caso na pré-campanha das eleições presidenciais brasileiras de 2018. **Brazilian Journalism Research**, v. 15, n. 3, p. 598–625, 2019.

CARNEVALI, J. B. L. **Evaluation Measures in Information Retrieval**. Pinecone, 2022. Disponível em: <https://www.pinecone.io/learn/offline-evaluation>. Acesso em: 31 mai. 2023.

CORDEIRO, P. R.; PINHEIRO, V. Um corpus de notícias falsas do twitter e verificação automática de rumores em língua portuguesa. In: **STIL-Brazilian Symposium in Information and Human Language Technology. IEEE, Salvador, BA, Brazil**. [S. l.: s. n.], 2019. p. 220–228.

EGELHOFER, J.; LECHELER, S. Fake news as a two-dimensional phenomenon: a framework and research agenda. **Annals of the International Communication Association**, v. 43, p. 1–20, 04 2019.

FISCHER, G. User modeling in human–computer interaction. **User modeling and user-adapted interaction**, Springer, v. 11, n. 1, p. 65–86, 2001.

FLORIDI, L. What is the philosophy of information? **Metaphilosophy**, Wiley Online Library, v. 33, n. 1-2, p. 123–145, 2002.

GOMEZ-URIBE, C. A.; HUNT, N. The netflix recommender system: Algorithms, business value, and innovation. **ACM Trans. Manage. Inf. Syst.**, Association for Computing Machinery, New York, NY, USA, v. 6, n. 4, dec 2016. ISSN 2158-656X. Disponível em: <https://doi.org/10.1145/2843948>.

GRBOVIC, N. **Why A Recommendation Engine for E-Commerce Boosts Sales**. Hyperon, 2023. Disponível em: <https://www.hyperon.io/blog/why-a-recommendation-engine-for-e-commerce-boosts-sales>. Acesso em: 30 mai. 2023.

HARRIS, Z. S. Distributional structure. **WORD**, Routledge, v. 10, n. 2-3, p. 146–162, 1954. Disponível em: <https://doi.org/10.1080/00437956.1954.11659520>.

HUNT, E. **What is fake news? How to spot it and what you can do to stop it**. The Guardian, 2016. Disponível em: <https://www.theguardian.com/media/2016/dec/18/what-is-fake-news-pizzagate>. Acesso em: 6 nov. 2022.

JACCARD, P. The distribution of the flora in the alpine zone. 1. **New phytologist**, Wiley Online Library, v. 11, n. 2, p. 37–50, 1912.

JHA, A. **Vectorization Techniques in NLP**. MLOps Blog, 2022. Disponível em: <https://neptune.ai/blog/vectorization-techniques-in-nlp-guide>. Acesso em: 4 dez. 2022.

JING F. LI, B. S. Z. Z. J.; CHOO, K. K. R. Disinformation propagation trend analysis and identification based on social situation analytics and multilevel attention network. **IEEE Transactions on Computational Social Systems**, 2022.

KULKARNI, N. **IR Presentation on Collaborative Filtering**. Slideshare, 2017. Disponível em: <https://www.slideshare.net/15koolneha/collaborative-filtering-73051273>. Acesso em: 15 nov. 2022.

KUMAR, L. K. **NLP: Bag of words and TF-IDF explained!** LinkedIn, 2021. Disponível em: <https://www.linkedin.com/pulse/nlp-bag-words-tf-idf-explained-l-koushik-kumar>. Acesso em: 30 mai. 2023.

- KUNEŠ, J. **Similarity and modeling in science and engineering**. [S. l.]: Springer Science & Business Media, 2012.
- LEWANDOWSKY, S.; ECKER, U. K.; COOK, J. Beyond misinformation: Understanding and coping with the “post-truth” era. **Journal of applied research in memory and cognition**, Elsevier, v. 6, n. 4, p. 353–369, 2017.
- LOPS, P.; GEMMIS, M. d.; SEMERARO, G. Content-based recommender systems: State of the art and trends. **Recommender systems handbook**, Springer, p. 73–105, 2011.
- LU, C.-C.; TSENG, V. S. A novel method for personalized music recommendation. **Expert Systems with Applications**, Elsevier, v. 36, n. 6, p. 10035–10044, 2009.
- MAO, L. **Cosine Similarity vs. Pearson Correlation Coefficient**. Lei Mao Blog / GitHub, 2021. Disponível em: <https://leimao.github.io/blog/Cosine-Similarity-VS-Pearson-Correlation-Coefficient>. Acesso em: 26 nov. 2022.
- MCFEE, B.; BARRINGTON, L.; LANCKRIET, G. Learning content similarity for music recommendation. **IEEE transactions on audio, speech, and language processing**, IEEE, v. 20, n. 8, p. 2207–2218, 2012.
- MERRIAM-WEBSTER. **The real story of ‘fake news’: The term seems to have emerged around the end of the 19th century**. Merriam-Webster, 2017. Disponível em: <https://www.merriam-webster.com/words-at-play/the-real-story-of-fake-news>. Acesso em: 6 nov. 2022.
- MONTEIRO, R. A.; SANTOS, R. L.; PARDO, T. A.; ALMEIDA, T. A. d.; RUIZ, E. E.; VALE, O. A. Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In: SPRINGER. **International Conference on Computational Processing of the Portuguese Language**. [S. l.], 2018. p. 324–334.
- NA, S.; XUMIN, L.; YONG, G. Research on k-means clustering algorithm: An improved k-means clustering algorithm. In: IEEE. **2010 Third International Symposium on intelligent information technology and security informatics**. [S. l.], 2010. p. 63–67.
- NARGESIAN, F.; SAMULOWITZ, H.; KHURANA, U.; KHALIL, E. B.; TURAGA, D. S. Learning feature engineering for classification. In: **Ijcai**. [S. l.: s. n.], 2017. p. 2529–2535.
- NEWMAN, N.; FLETCHER, R.; SCHULZ, A.; ANDI, S.; ROBERTSON, C. T.; NIELSEN, R. K. Reuters institute digital news report 2021. **Reuters Institute for the study of Journalism**, 2021.
- O’DONNELL, V.; JOWETT, G. S. **Propaganda and persuasion**. [S. l.]: Sage, 2004. v. 6. 7 p.
- O’NEILL, B. Frame fields. In: **Elementary Differential Geometry**. [S. l.]: Academic Press Boston, 2006. p. 43–99.
- PARZEN, E. On Estimation of a Probability Density Function and Mode. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 33, n. 3, p. 1065 – 1076, 1962. Disponível em: <https://doi.org/10.1214/aoms/1177704472>.
- PATANKAR, A.; BOSE, J.; KHANNA, H. A bias aware news recommendation system. In: **2019 IEEE 13th International Conference on Semantic Computing (ICSC)**. [S. l.: s. n.], 2019. p. 232–238.

- PATRA, B. K.; LAUNONEN, R.; OLLIKAINEN, V.; NANDI, S. A new similarity measure using bhattacharyya coefficient for collaborative filtering in sparse data. **Knowledge-Based Systems**, Elsevier, v. 82, p. 163–177, 2015.
- PEARSON, K. Vii. note on regression and inheritance in the case of two parents. **proceedings of the royal society of London**, The Royal Society London, v. 58, n. 347-352, p. 240–242, 1895.
- REIS, R.; ZANETTI, D.; FRIZZERA, L. Algoritmos e desinformação: O papel do youtube no cenário político brasileiro. In: **VIII Congresso da Associação Brasileira de Pesquisadores em Comunicação e Política**. [S. l.: s. n.], 2019.
- RESENDE, G.; MELO, P.; SOUSA, H.; MESSIAS, J.; VASCONCELOS, M.; ALMEIDA, J.; BENEVENUTO, F. (mis) information dissemination in whatsapp: Gathering, analyzing and countermeasures. In: **The World Wide Web Conference**. [S. l.: s. n.], 2019. p. 818–828.
- RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender systems: Techniques, applications, and challenges. In: \_\_\_\_\_. **Recommender Systems Handbook**. New York, NY: Springer US, 2022. p. 1–35. ISBN 978-1-0716-2197-4. Disponível em: [https://doi.org/10.1007/978-1-0716-2197-4\\_1](https://doi.org/10.1007/978-1-0716-2197-4_1).
- ROCCA, B. **Introduction to recommender systems**. Towards Data Science, 2019. Disponível em: <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>. Acesso em: 19 nov. 2022.
- ROSENBLATT, M. Remarks on Some Nonparametric Estimates of a Density Function. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 27, n. 3, p. 832 – 837, 1956. Disponível em: <https://doi.org/10.1214/aoms/1177728190>.
- RUIZ C., . N. T. D. Disinformation and echo chambers: How disinformation circulates on social media through identity-driven controversies. **Journal of Public Policy Marketing**, 2022.
- SALLAMI, D.; SALEM, R. B.; AÏMEUR, E. Trust-based recommender system for fake news mitigation. In: **Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization**. New York, NY, USA: Association for Computing Machinery, 2023. (UMAP '23 Adjunct), p. 104–109. ISBN 9781450398916. Disponível em: <https://doi.org/10.1145/3563359.3597395>.
- SCHLESINGER, R. **Fake News in Reality**. U.S. News and World Report, 2017. Disponível em: <https://www.usnews.com/opinion/thomas-jefferson-street/articles/2017-04-14/what-is-fake-news-maybe-not-what-you-think>. Acesso em: 6 nov. 2022.
- SKOVHØJ, F. Z. **Using Collaborative Filtering in E-Commerce: Advantages Disadvantages**. Clerk.io Blog, 2022. Disponível em: <https://blog.clerk.io/collaborative-filtering>. Acesso em: 21 nov. 2022.
- SOLL, J. **The Long and Brutal History of Fake News**. Politico Magazine, 2016. Disponível em: <https://www.merriam-webster.com/words-at-play/the-real-story-of-fake-news>. Acesso em: 6 nov. 2022.
- STITINI, O.; KALOUN, S.; BENCHAREF, O. Towards the detection of fake news on social networks contributing to the improvement of trust and transparency in recommendation systems: Trends and challenges. **Information**, v. 13, n. 3, 2022. ISSN 2078-2489. Disponível em: <https://www.mdpi.com/2078-2489/13/3/128>.

TIEMANN, M.; PAUWS, S. Towards ensemble learning for hybrid music recommendation. In: **Proceedings of the 2007 ACM conference on Recommender systems**. [S. l.: s. n.], 2007. p. 177–178.

TREVISO, M.; JI, T.; LEE, J.-U.; AKEN, B. van; CAO, Q.; CIOSICI, M. R.; HASSID, M.; HEAFIELD, K.; HOOKER, S.; MARTINS, P. H.; MARTINS, A. F. T.; MILDER, P.; RAFFEL, C.; SIMPSON, E.; SLONIM, N.; BALASUBRAMANIAN, N.; DERCZYNSKI, L.; SCHWARTZ, R. **Efficient Methods for Natural Language Processing: A Survey**. arXiv, 2022. Disponível em: <https://arxiv.org/abs/2209.00099>.

VERMEULE, C. A.; SUNSTEIN, C. R. Conspiracy theories: causes and cures. **Journal of Political Philosophy**, Wiley-Blackwell, p. 205, 2009.

WANG, S.; XU, X.; ZHANG, X.; WANG, Y.; SONG, W. Veracity-aware and event-driven personalized news recommendation for fake news mitigation. In: **Proceedings of the ACM Web Conference 2022**. New York, NY, USA: Association for Computing Machinery, 2022. (WWW '22), p. 3673–3684. ISBN 9781450390965. Disponível em: <https://doi.org/10.1145/3485447.3512263>.

WOOLLEY, S. C.; HOWARD, P. N. Political communication, computational propaganda, and autonomous agents. *International Journal of Communication*, 2016.

YOCKEY, H. P. **Information Theory, Evolution, and the Origin of Life**. [S. l.]: Cambridge University Press, 2005. v. 2005. 7 p. ISBN 978-0-511-54643-3.

ZAROCOSTAS, J. How to fight an infodemic. **The lancet**, Elsevier, v. 395, n. 10225, p. 676, 2020.