UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA E DE TELECOMUNICAÇÕES

NICOLLAS RODRIGUES DE OLIVEIRA

Aplicações Eficientes do Processamento de Linguagem Natural: Identificação de Notícias Falsas, Sumarização Automática de Textos e Construção de Ontologias

NITERÓI

2020

UNIVERSIDADE FEDERAL FLUMINENSE
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA E DE TELECOMUNICAÇÕES

NICOLLAS RODRIGUES DE OLIVEIRA

Aplicações Eficientes do Processamento de Linguagem Natural: Identificação de Notícias Falsas, Sumarização Automática de Textos e Construção de Ontologias

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Telecomunicações da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica e de Telecomunicações. Área de concentração: Comunicação de Dados Multimídia.

Orientador:
Diogo Menezes Ferrazani Mattos

NITERÓI

2020

Ficha catalográfica elaborada pelo Sistema de Bibliotecas da UFF - SDC/UFF
com os dados fornecidos pelo(a) autor(a)

S586t Silva Junior, José Lins da
Título do Trabalho: subtítulo do trabalho / José Lins da Silva Junior; Orientador Sobrenome, orientador; Coorientador Sobrenome, co-orientador. Niterói, 2017.
120 f.

Monografia (Especialização em Engenharia de Produção) - Universidade Federal Fluminense, Laboratório de Tecnologia, Gestão de Negócios e Meio Ambiente, Niterói, 2017.

1. Primeiro assunto. 2. Produção intelectual. I. Título II. Sobrenome,Orientador, orientador. III. Sobrenome, Coorientador, co-orientador. IV. Universidade Federal Fluminense. Laboratório de Tecnologia, Gestão de Negócios e Meio Ambiente.

CDD -

NICOLLAS RODRIGUES DE OLIVEIRA

Aplicações Eficientes do Processamento de Linguagem Natural: Identificação de Notícias Falsas, Sumarização Automática de Textos e Construção de Ontologias

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Telecomunicações da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica e de Telecomunicações. Área de concentração: Comunicação de Dados Multimídia.

Aprovada em 15 de Setembro de 2020.

BANCA EXAMINADORA

Prof. Diogo M. F. Mattos, D.Sc. – Orientador, UFF

Prof$^{\underline{a}}$. Dianne Scherly Varela de Medeiros, D.Sc. – UFF

Prof. Otto Carlos Muniz Bandeira Duarte, Dr. Ing. – UFRJ

Niterói

2020

A mim mesmo

Agradecimentos

Gostaria de agradecer primeiramente a mim mesmo por ter me acompanhado durante toda minha trajetória na graduação, no mestrado e acredito que na vida também. Sem você, Nicollas esta dissertação não seria concluída.

Em segundo lugar, gostaria de agradecer a parte mais importante da minha família – mamãe, vovó, padrasto e irmãozinho – que, acredito eu, sabem que eu não fiz mestrado em telemarketing.

Gostaria de agradecer ao meu companheiro Eric, pelos inúmeros "Calma", que me motivaram e tranquilizaram nos momentos de crise existencial pré-*deadlines*.

Agradeço imensamente ao meu orientador, Professor Diogo M. F. Mattos, por todo o suporte motivacional e teórico essenciais para completar essa jornada.

Meus sinceros agradecimentos à Professora Dianne S. V. Medeiros, por todo apoio e por tentar me ensinar a sábia arte da tolerância com os chatos, embora eu insista em não absorver.

Minha gratidão a todos professores, colaboradores e alunos do Laboratório Mídiacom pelo conhecimento compartilhado e amizade construída.

Por fim, agradeço aos órgãos de fomento CNPq, CAPES, RNP e FAPERJ.

Resumo

A disponibilidade crescente de dados não é sinônimo de informação útil e muito menos de conhecimento. Essa não-linearidade é corroborada pelo conceito de "infodemia", que expressa o excesso de informações, algumas precisas e outras não, em torno de um assunto, dificultando assim encontrar fontes idôneas e orientações confiáveis. Tal conceito está intimamente relacionado com às notícias falsas e seus efeitos negativos. Além de confundir o leitor, a divulgação de conteúdo falso também implica desperdícios de recursos da rede, processamento e consiste em grave ameça à integridade das informações e à credibilidade do serviço prestado. Nesse contexto, a tarefa de extração de conhecimento a partir de dados não estruturados torna-se ainda mais complexa. Esta dissertação propõe desenvolver três soluções, eficientemente embasadas no processamento de linguagem natural, para atuar na sumarização, representação e classificação de textos. Assim, primeiramente focando na condensação de informações textuais, propõe-se a Rezzumin, uma ferramenta *web* capaz de resumir automaticamente textos em português com base na extração de características de grafos de conhecimento. Além do uso de um tesauro (dicionário de sinônimos) na normalização léxica, a ferramenta pondera concisão e diversidade de tópicos no resumo gerado por meio da aplicação de métricas de centralidade e do algoritmo *k-medoids*. Os resultados da avaliação mostram alta precisão e abrangência na síntese dos documentos científicos ao comparar o resumo gerado computacionalmente e o resumo fornecido por autores de artigos. Paralelamente, propõe-se um nova abordagem de relacionamento sintático para geração automática de taxonomias de conhecimento sobre textos de domínio específico. A abordagem considera um estudo de caso de computação em nuvem por meio da coleta e análise de um conjunto de publicações científicas recentes. Para avaliar esta proposta, foi conduzido uma comparação quantitativa entre diferentes métricas intrínsecas de agrupamento. Os resultados apontam maior popularidade e cobertura da presente proposta do que o estado da arte, especialmente quando se utiliza agrupamento hierárquico. O diferencial está na construção de uma representação bem informativa do conhecimento com apenas três quartos dos dados textuais originais e sem qualquer rotulação de verdade fundamental. Por último é proposto também uma análise estilístico-computacional com a finalidade de detectar notícias falsas em textos extraídos de mídias sociais. A análise considera notícias do Twitter, das quais foram coletados aproximadamente 33.000 *tweets*, classificados entre reais e comprovadamente falsos. Na avaliação da qualidade da detecção, os resultados encontram valores expressivos de precisão e sensibilidade mesmo empregando uma redução dimensional para um sexto do número de características originais. Esta última proposta detém uma sobrecarga mínima, embora tenha o potencial de fornecer um alto índice de confiança na diferenciação de notícias falsas de verdadeiras.

Palavras-chave: Processamento de Linguagem Natural, Sumarização Automática, Ontologias, Detecção de Notícias Falsas.

Abstract

The increasing availability of data is not synonymous with useful information, much less knowledge. This non-linearity is corroborated by the concept of "infodemic", which expresses the excess of information, some accurate and others not, around a subject, making it challenging to find suitable sources and reliable guidance. Such a concept is closely related to false news and its harmful effects. In addition to confusing the reader, the dissemination of false content also implies a waste of network resources, processing, and a serious threat to the integrity of the information and the credibility of the service provided. In this context, the task of extracting knowledge from unstructured data becomes even more complex. This dissertation proposes to develop three solutions, efficiently based on natural language processing, to act in the summarization, representation, and classification of texts. Thus, first focusing on the condensation of textual information, Rezzumin is proposed, a web tool capable of automatically summarizing texts in Portuguese based on the extraction of knowledge graph resources. In addition to using a thesaurus (synonyms dictionary) in lexical normalization, the tool weighs conciseness and diversity of topics in summary generated by applying centrality metrics and the k-medoids algorithm. The evaluation results show high precision and comprehensiveness in summarizing scientific documents when comparing the computationally generated abstract and the abstract provided by authors of articles. Simultaneously, a new syntactic relationship approach is proposed for the automatic generation of knowledge taxonomies on specific domain texts. The approach considers a case study of cloud computing by collecting and analyzing a set of recent scientific publications. In order to evaluate this proposal, a quantitative comparison between different intrinsic cluster metrics was conducted. The results point to greater popularity and coverage of the present proposal than state of the art, especially when using hierarchical grouping. The difference lies in constructing a very informative representation of knowledge with only three-quarters of the original textual data and without any fundamental truth labeling. Finally, a stylistic-computational analysis is also proposed to detect false news in texts extracted from social media. The analysis considers news from Twitter, of which approximately 33,000 tweets were collected, classified between real and proven to be false. In evaluating the quality of the detection, the results find expressive values of precision and sensitivity even when employing a dimensional reduction to one-sixth of the number of original resources. This latter proposal has minimal overhead, although it can provide a high level of confidence in differentiating between false and authentic news.

Keywords: Natural Language Processing, Automatic Summarization, Ontologies, Fake News Detection

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

PLN	Processamento de Linguagem Natural
TF-IDF	Frequência do Termo – Inverso da Frequência nos Documentos
BoW	Saco-de-Palavras
LSI	Indexação Semântica Latente
LSA	Análise Semântica Latente
SVM	Máquina de Vetor de Suporte
RBF	Radial Basis Function
RF	Floresta Aleatória
kNN	k-Vizinhos Mais Próximos
DBSCAN	Clusterização Espacial Baseada em Densidade de Aplicações com Ruído
OCR	Reconhecimento Óptico de Caracteres
NGD	Normalized Google Distance

Sumário

Capítulo 1

Introdução

Quando não ordenada e organizada, a informação perde seu valor fundamental, tornando-se apenas um dado sem qualquer sentido adicional. Em 2013, apenas 22% das informações digitais eram passíveis de análise, sendo capazes de servir como matéria-prima para geração de metadados e para a extração de conhecimento útil. Todavia, menos de 5% de tais informações foram efetivamente analisadas. Espera-se que ao final 2020 o percentual útil alcance 35% dos aproximadamente 44 zettabytes de dados, um reflexo direto do crescimento de dados dos sistemas embarcados [1]. A qualidade dos dados impacta na sua inteligibilidade, sobretudo quando armazenados na forma de textos escritos em linguagem natural, modo cotidiano de comunicação entre humanos. A presença de ruídos textuais indesejáveis como abreviações, erros de ortografia e de concordância e gírias, pode comprometer a navegação, organização, busca e interpretação inteligente da informação.

Atualmente, toda comunidade acadêmica é diariamente motivada a investigar, analisar e posteriormente publicar avanços científicos que possam auxiliar no combate ao Covid-19. Ao documentar o estado da arte de um campo de interesse da tecnologia, as publicações científicas são fontes genuínas de informação para aquisição e compreensão de conhecimento. Na tentativa de expandir o acesso ao conhecimento, os métodos bibliométricos cumprem um papel essencial de extrair padrões e tendências de forma eficiente de patentes, periódicos e repositórios de conferências. A premissa de que a complexidade do processamento e análise textual é inversamente proporcional à estruturação das fontes de dados utilizadas, coloca fontes estruturadas ou semiestruturadas, como esquemas de banco de dados e dicionários, respectivamente, em uma condição muito mais permissível para manipulação. Apesar do potencial de conhecimento embutido nesses documentos

[1] Disponível em: https://www.emc.com/leadership/digital-universe/ 2014iview/executive-summary.html.

acadêmicos, seja em arquivos de texto ou em páginas da *web*, eles ainda são fontes de texto não estruturadas escritas em linguagem natural. Além de possuírem essa densidade maior de informações úteis, as fontes não estruturadas também são estatisticamente mais abundantes. No final de 2020, IDC [2] estima que os dados não estruturados representem 95% dos dados globais, com uma taxa de crescimento anual estimada de 65%. Diante dessas estatísticas e predições alarmantes a respeito do volume de dados não estruturados, destaca-se três desafios principais: resumos automáticos, ontologias e *fake news*.

O primeiro envolve a necessidade de extração de informações relevantes de textos. O volume de documentos eletrônicos disponíveis leva os usuários a enfrentarem dificuldades para encontrarem informações relevantes inéditas, dado o alto nível de redundância dos documentos que encobrem informações relevantes porém pouco prevalentes. Nesse sentido, a condensação de informações importantes na forma de resumo propicia que os usuários tenham acesso rápido às informações relevantes sem que se deparem com o excesso de informação redundante dos documentos eletrônicos. Assim, um desafio atual de pesquisa é desenvolver novas ferramentas para resumir automaticamente textos de propósito geral.

A geração de resumos automáticos é uma ferramenta importante e eficiente no auxílio à interpretação, já que há uma limitação da capacidade humana de resumir mentalmente uma grande quantidade de texto [2]. A geração de resumos automático de textos de propósito geral consiste em um problema de otimização de múltiplos objetivos que visa procurar trechos relevantes em meio a um grande volume de informações disponíveis e, ao mesmo tempo, absorver uma grande quantidade de informações, maximizando o número de tópicos cobertos pelo resumo. O objetivo final do resumo automático é condensar o texto original em uma versão mais curta, preservando o conteúdo das informações relevantes e o seu significado geral [3].

O segundo desafio remete a ideia de que a capacidade de armazenar e acessar informações de forma eficiente é tão importante quanto ser capaz de extrair tais informações. A aplicabilidade dessa afirmação foi verificada nos anos 90, quando a migração em larga escala do conteúdo textual da mídia física para a *web* popularizou mecanismos de busca como o Altavista. Essa transferência de informação, junto com o aumento da demanda por ontologias, que promoveu a *Web* Semântica, demarcou a construção automática de ontologias, o intitulado aprendizado de ontologias, como objeto de pesquisa. O aprendizado de ontologia tem como objetivo identificar e extrair instâncias textuais relevantes de um silo cada vez maior de informações em construções compartilháveis de alto nível para

[2] Disponível em: https://www.emc.com/leadership/digital-universe/2012iview/cloud-computing-in-2020.htm

melhorar as aplicações cotidianas [4, 5, 6].

Soluções manuais ou mesmo cooperativas para o desenvolvimento de ontologias requerem supervisão especializada. Total ou parcial, o resultado é um processo demorado, extremamente trabalhoso e caro. No extremo oposto, as abordagens totalmente automáticas, conforme previsto em um esquema de aprendizado de ontologia teórica, ainda são uma lacuna, que possivelmente não será preenchida em um futuro próximo. A alternativa restante é a semi-automatização do processo, prevendo uma intervenção humana mínima, mas ainda existente. Além de suprimir a participação humana, tais abordagens precisam lidar com fontes de dados não estruturados, o que é um complicador nos casos em que a identificação de termos mais importantes é latente e depende de um processamento linguístico bem desenvolvido [7, 8, 9]

O terceiro desafio relaciona-se com a identificação e classificação de texto, sobretudo aqueles com caráter danoso como é o caso de notícias falsas (*fake news*). O combate às notícias falsas torna indissociáveis os problemas de integridade e veracidade da informação em rede social e do consumo de dados na camada de aplicação. Dessa forma, o compartilhamento de informações inverídicas diz respeito à qualidade da confiança (*Quality of Trust* - QoT) aplicada à distribuição de notícias [10], referindo-se ao quanto um usuário confia em um conteúdo de uma determinada fonte. Em diferentes países, observam-se baixos níveis de confiança nas mídias de massa, *e.g.* apenas 40% nos Estados Unidos[3], enquanto há altas porcentagens de compartilhamento de *links* nunca lidos (*blindshares*), *e.g.* 59% no Reino Unido [11].

O aumento da disseminação das notícias falsas é resultado da expansão do uso de redes sociais que agilizam a propagação de boatos, sátiras e informações erradas. É difícil para um indivíduo diferenciar entre o que é verdadeiro e o que é falso enquanto é sobrecarregado com informações enganosas que são recebidas por repetidas vezes. Ademais, os indivíduos tendem a confiar em notícias falsas porque há atualmente uma descrença do público em relação às mídias tradicionais e, porque, muitas vezes tais notícias são compartilhadas por amigos ou confirmam um conhecimento prévio [12]. Isso torna a identificação de notícias falsas mais crítica em comparação com outros tipos de informações, já que geralmente são apresentadas com elementos que lhe conferem autenticidade e objetividade, sendo relativamente mais fácil de obter a confiança do público.

[3]Disponível em: https://news.gallup.com/poll/185927/americans-trust-media-remains-historical-low.aspx

1.1 Motivação

A qualidade dos dados é um desafio para a extração de informação útil. Um exemplo é a governança de cidades inteligentes, como visto em Volta Redonda, em que o banco de dados do Centro Integrado de Operações de Segurança Pública (CIOSP) é mantido. Tal banco é alimentado com informações coletadas em ligações recebidas diariamente pela central de atendimento à população. Cada atendimento, dito ocorrência, representa uma solicitação a um dos órgãos de segurança integrados ao centro, como guarda municipal, polícia militar, polícia civil, corpo de bombeiros, defesa civil e central de ambulâncias. Ao atender o solicitante da ligação, os vários atendentes registram no banco de dados uma breve descrição do que foi a ocorrência. Apesar de já possuir um reflexo positivo no direcionamento dos recursos públicos no atendimento da população local, essa iniciativa ainda se encontra subutilizada. Em especial, a base de dados do registro de ocorrências armazena dados importantes para o entendimento da segurança pública na cidade, porém os dados são desestruturados e isolados em um silo de dados não processados.

Contudo, análises preliminares do banco de dados mostram que, embora estruturados em tabelas, os dados armazenados, cerca de 500 mil ocorrências, não detêm qualquer tipo de tratamento textual, padronização ou classificação. Nesse estado, torna-se difícil a completa extração de conhecimento do banco, podendo mascarar estatísticas importantes para os órgãos relacionados. Além disso, lidar com a aleatoriedade dos dados de entrada implica o aumento da complexidade das regras empregadas na identificação de padrões na descrição de cada ocorrência. Após a aplicação de técnicas de processamento de processamento de linguagem natural combinadas com algoritmos de aprendizado de máquina, foi possível extrair informações até então desconhecidas para as autoridades do município. Resultados evidenciaram forte incidência de ocorrências requisitando policiais, além de quantidades significativas de ocorrências relacionadas à poluição sonora e estacionamento irregular [13].

1.2 Contribuição

Esta dissertação contém três contribuições principais, sendo a primeira delas o desenvolvimento de um protótipo de ferramenta *web* híbrida capaz de fornecer automaticamente resumos extrativos, com alto grau de informações relevantes, a partir de textos em português inseridos pelos usuários. A segunda contribuição desta dissertação, é uma abordagem baseada no relacionamento sintático que permite a geração automática de estruturas

ontológicas a partir de um conjunto de publicações científicas. A terceira contribuição concentra-se em uma análise estilístico-computacional que implementa uma metodologia tripla na detecção de notícias falsas extraídas do *Twitter*. Essas metodologias variam desde abordagens usando combinações de algoritmos de aprendizado de máquina até interpretações estatísticas.

1.3 Organização da Dissertação

Esta dissertação está organizada da seguinte forma. O Capítulo 2 apresenta uma breve fundamentação teórica, revisitando conceitos de processamento textual e vetorial, bem como algoritmos de aprendizado de máquina e suas formas de avaliação. O Capítulo 3 relata os trabalhos relacionados ao processamento de linguagem natural e suas aplicações. O Capítulo 4 expõe e avalia a ferramenta proposta para sumarização automática de textos. O Capítulo 5 apresenta uma abordagem híbrida para construção de estruturas hierárquicas de conhecimento a partir de textos de um domínio específico. O Capítulo 6 propõe uma metodologia tripla para detecção de notícias falsas baseada na análise estilístico-computacional de textos originários de mídias sociais. Finalmente, o Capítulo 7 conclui a dissertação.

Capítulo 2

Fundamentação Teórica

Esta seção apresenta o embasamento teórico que irá nutrir as três aplicações abordadas na dissertação. Dessa forma são revisitados os principais procedimentos e técnicas usadas no processamento e representação vetorial eficiente de dados de origem textual. Também são abordados algoritmos de classificação e agrupamento, métricas de semelhança e, por fim, formas de avaliação com ou sem dados de referência.

2.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN), também conhecido como linguística computacional, consolida-se como uma campo de pesquisa que envolve modelos e processos computacionais para a solução de problemas práticos de compreensão e manipulação de linguagens humanas. Independentemente sua forma de manifestação, textual ou fala, a linguagem natural é entendida como qualquer forma de comunicação diária entre humanos. Tal definição exclui linguagens de programação e notações matemáticas, consideradas linguagens artificiais, uma vez que as linguagens naturais estão em constante mudança, dificultando o estabelecimento de regras explícitas para computadores [14, 15, 16].

Em uma decomposição refinada, o PLN pode ser dividido em cinco estágios primários de análise, que, quando realizados, permitem que o significado pretendido pelo autor seja extraído computacionalmente de um documento textual. Embora seja mais condizente com um estágio de pré-processamento, o primeiro estágio é a segmentação por *tokenização*. A *tokenização* é uma técnica obrigatória dado que os documentos textuais em linguagem natural geralmente são compostos de frases longas, complicadas e mal formadas. A etapa seguinte é a análise léxica, que visa relacionar as variantes morfológicas aos seus *lemas*, ou seja, a forma primitiva das palavras do dicionário. A análise sintática foca no relaci-

onamento das palavras entre si, cada uma assumindo seu papel estrutural nas frases, e de como as frases podem ser partes de outras, constituindo sentenças. Linguisticamente, a análise semântica tenta destilar o significado de palavras, expressões fixas, sentenças inteiras, sendo assim frequentemente aplicada na resolução de ambiguidades. Por fim, a análise pragmática busca compreender uma determinada frase, observando referências pronomiais e a coerência textual da estrutura das frases adjacentes. Embora o PLN possa introduzir outros estágios de análise, como reconhecimento de emoção, esses cinco estágios básicos são suficientes para extrair a informação semântica contextualizada de um documento de linguagem natural.

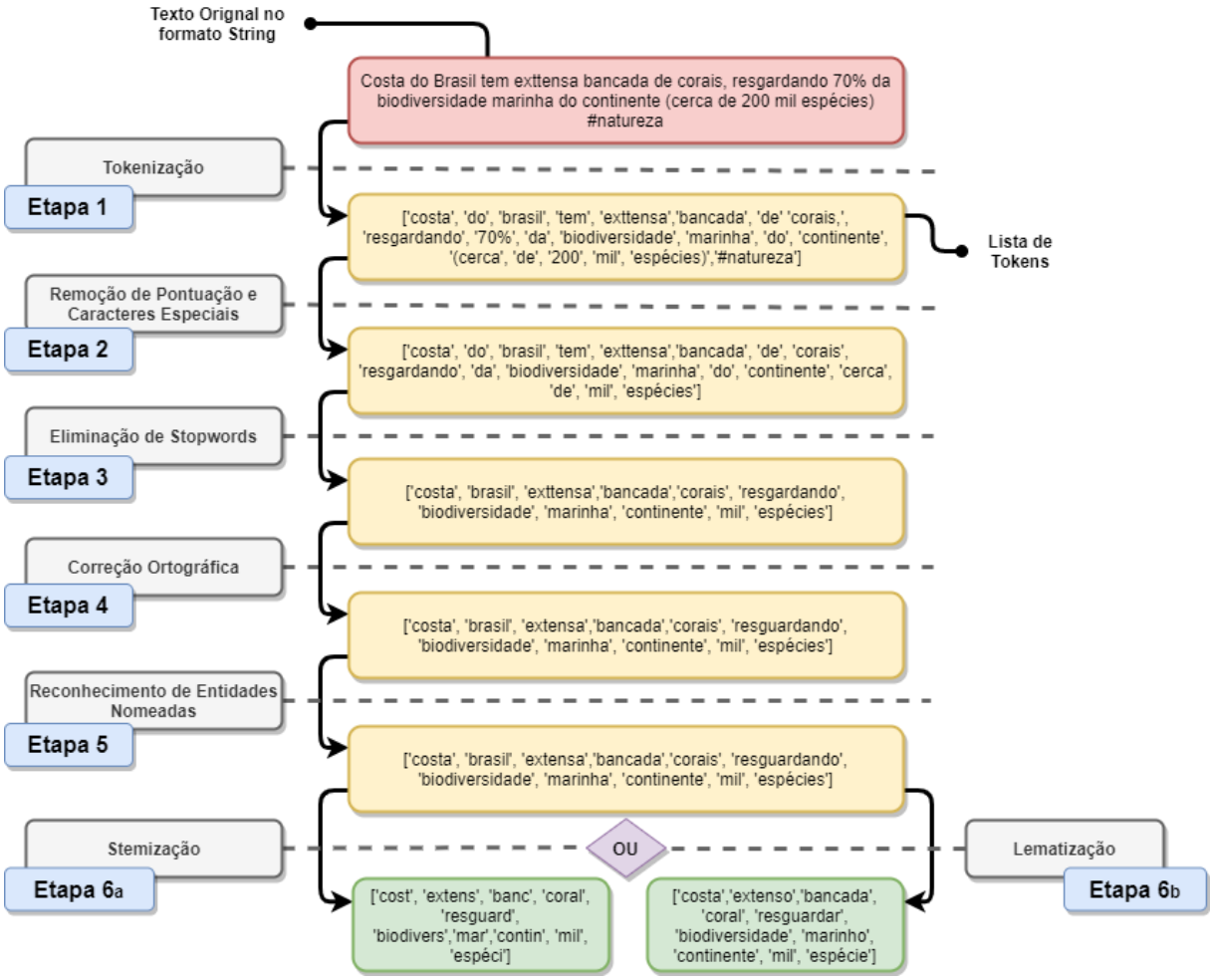


Figura 2.1: Fluxograma genérico para aplicação do processamento de linguagem natural em uma sentença.

Limitando o processamento até o estágio de análise morfológica, é possível compor uma sequência básica de técnicas de PLN para garantir a identificação, e posterior remoção, de qualquer ruído textual que possa comprometer a extração e interpretação inteligente das informações contidas em cada sentença. Nesta sequência, ilustrada na Figura 2.1, são aplicadas técnicas de limpeza e conformação dos dados incluindo *tokenização*,

remoção de pontuação e caracteres especiais, eliminação de *stopwords*, correção ortográfica, reconhecimento de entidades nomeadas e *stemização* ou *lematização*. Guiada pela ordem acima mencionada, cada *sentença* do texto original é primeiramente submetida a um procedimento de discretização visto na Etapa 1, conhecido como *tokenização*. Usando neste caso o caractere de espaço como critério delimitador, a *tokenização* transforma cada sentença contígua em uma lista de *tokens*, permitindo o manuseio individualizado dos *tokens*. Basicamente cada *token* é visto como uma instância de uma sequência de caracteres. Posteriormente na Etapa 2, recursos ortográficos como pontuação, *e.g.* pontos final, de exclamação e de interrogação, e caracteres especiais, *e.g.* números, cifrão e asterisco, são removidos de cada *token*. Na Etapa 3, eliminam-se as *stopwords*, ou palavras mais frequentes, como conectivos, artigos e pronomes. Essa tarefa em especial tem como base o princípio de que quanto maior a frequência de uma palavra no *corpus*, menos informação relevante a palavra possui. Em seguida na Etapa 4, ocorre a correção ortográfica através da comparação do *token* com seu correspondente mais próximo no dicionário. Executa-se tal procedimento calculando a distância Levenshtein, *i.e.*, o número mínimo de operações necessárias para transformar um nome no banco de dados em outro contido em um dicionário de nomes. O reconhecimento de entidades nomeadas, Etapa 5, identifica principalmente nomes próprios, com subsequente remoção dessas palavras. Na *stemização*, as palavras flexionadas ou derivadas são reduzidas ao seu radical, eliminando possíveis variantes ou plurais. Por fim, com o objetivo de reduzir o processamento desnecessário causado por eventuais redundâncias entre as palavras, seja por flexões ou derivações, é comum a adoção da Etapa 6a ou 6b, sendo respetivamente a *lematização* e a *stemização*. Na tarefa de *lematização*, procura-se eliminar as possíveis variantes ou plurais de uma mesma palavra, reduzindo-as ao mesmo lemas, conhecidos como forma de dicionário. Em contrapartida, na *stemização* esta redução é feita transformando cada palavra no seu radical [13, 17, 18].

2.2 Representação Vetorial de Elementos Textuais

Mesmo devidamente padronizada, cada sentença não é passível de ser operada matematicamente, visto que ainda é composta por radicais de palavras e não por valores mensuráveis. Para obter uma representação numérica, emprega-se o modelo de espaço vetorial. Este modelo define que textos, sejam sentenças ou documentos, podem ser interpretados como um espaço vetorial de palavras, em que cada palavra pode ser representada em diferentes padrões, tais como: o binário, Saco-de-Palavras, Frequência do Termo –

Inverso da Frequência nos Documentos (*Term Frequency–Inverse Document Frequency*, TF-IDF). Para ilustrar as particularidades de cada padrão de vetorização, consideremos o *corpus* [1] da Tabela 2.1 formado por uma coletânea de quatro documentos, cada um contendo apenas uma única sentença. Devido a unicidade na quantidade de sentenças adotada no *corpus* exemplo, as descrições a seguir tem o objetivo de mostrar as possíveis representações vetoriais em nível de documento e não em nível de sentença, embora isto seja igualmente viável.

Tabela 2.1: *Corpus* exemplo

Documento 1 (D1)	Primeira sentença do corpus
Documento 2 (D2)	A segunda sentença é curta
Documento 3 (D3)	A terceira é curta
Documento 4 (D4)	A quarta sentença é a maior do corpus

2.2.1 Binário

Consiste no modelo mais intuitivo de vetorização, em que para cada palavra é atribuído um valor 1 ou 0 de acordo com sua presença ou ausência na sentença. Embora simples, é possível constatar pela Tabela 2.1 que este padrão de representação é pobre do ponto de vista semântico, uma vez que não traz qualquer informação sobre a importância de um termo para o conjunto de textos.

Tabela 2.2: Representação vetorial do *corpus* exemplo da Tabela 2.1 no modelo binário.

Termos	"primeira"	"quarta"	"a"	"corpus"	"curta"	"do"	"maior"	"segunda"	"sentença"	"terceira"	"é"
D1	1	0	0	1	0	1	0	0	1	0	0
D2	0	0	1	0	1	0	0	1	1	0	1
D3	0	0	1	0	1	0	0	0	0	1	1
D4	0	1	1	1	0	1	1	0	1	0	1

2.2.2 Saco-de-Palavras

O Saco-de-Palavras, tradução livre para *Bag-of-Words* (BoW), caracteriza-se como um tipo de modelo vetorial que atribui pesos aos termos, correspondentes ao número de ocorrências observadas do termos no texto. Matematicamente, os vetores dessa representação são expressos conforme a Equação 2.1, em que V_D é o vetor de pesos w para cada sentença do documento D até o n-iésimo termo.

$$V_D = [w_1, w_2, ..., w_{n-1}, w_n] \tag{2.1}$$

[1] Linguisticamente, um *corpus* é uma coletânea de documentos sobre determinado tema. Um conjunto de *corpus* é denominado *corpora*.

A Tabela 2.3 ressalta a presença de um peso igual a 2 na última linha da coluna referente ao termo "a". Isto de fato está condizente com a quantidade de vezes que esse termo aparece em D4 na Tabela 2.1, entretanto não reflete a importância semântica para o *corpus* considerado.

Tabela 2.3: Representação vetorial do *corpus* exemplo da Tabela 2.1 no modelo *Bag-of-Words*.

Termos	"primeira"	"quarta"	"a"	"corpus"	"curta"	"do"	"maior"	"segunda"	"sentença"	"terceira"	"é"
D1	1	0	0	1	0	1	0	0	1	0	0
D2	0	0	1	0	1	0	0	1	1	0	1
D3	0	0	1	0	1	0	0	0	0	1	1
D4	0	1	2	1	0	1	1	0	1	0	1

Este modelo de representação, assim como seu antecessor, sofrem do mesmo problema crítico, a presunção de uma igualdade de relevância de todos os termos perante ao *corpus*. Tal suposição pode conferir resultados questionáveis, um vez que, termos com alta ocorrência em um único documento podem eventualmente ser supervalorizados em uma avaliação baseada na soma total de cada termo no *corpus* [19].

2.2.3 Frequência do Termo – Inverso da Frequência nos Documentos

Esse modelo clássico de vetorização é definido na Equação 2.2 como o produto de duas medidas estatísticas, a **frequência do termo** (TF) e a **inverso da frequência nos documentos** (IDF). Embora o cálculo de frequência do termo (tf) siga a mesma lógica apresentada na Seção 2.2.2, o diferencial está na sua ponderação por idf_t, uma parcela que remete a quanto esse termo é citado nos demais documentos. Em sua fórmula, expressa na Equação 2.3, defini-se N como a contabilização do número de ocorrências do termo t no conjunto de documentos e df_t considera a frequência do termo t no documento em questão.

$$tfidf_t = tf_{t,d} \times idf_t \tag{2.2}$$

$$idf_t = log\frac{N}{df_t} \tag{2.3}$$

Essa modificação permite mensurar o grau de relevância semântica de um termo de um documento, em relação a toda coletânea. Como esperado, verifica-se que a Tabela 2.4 possui a mesma quantidade de linhas e colunas do modelo Saco-de-Palavras. Uma variante do TF-IDF original, é conhecido como TF-ISF (*Term Frequency – Inverse Sentence Frequency*), sendo largamente empregada na sumarização de textos em nível de sentença e não em nível de documento como o TF-IDF.

Tabela 2.4: Representação vetorial do *corpus* exemplo da Tabela 2.1 no modelo TF-IDF.

Termos	"primeira"	"quarta"	"a"	"corpus"	"curta"	"do"	"maior"	"segunda"	"sentença"	"terceira"	"é"
D1	0.614	0	0	0.484	0	0.484	0	0	0.392	0	0
D2	0	0	0.378	0	0.467	0	0	0.592	0.378	0	0.378
D3	0	0	0.408	0	0.505	0	0	0	0	0.640	0.408
D4	0	0.419	0.535	0.330	0	0.330	0.419	0	0.267	0	0.267

Um ponto importante a ser esclarecido é que, independente da representação aplicada, a dimensão do vetor está vinculada à quantidade restante de palavras distintas contidas em todo o banco de dados, já que várias delas foram removidas durante as etapas descritas na Seção 2.1. As palavras mantidas na sentença são as que carregam significado e, portanto, são as mais importantes para o entendimento da ideia central do texto. Ao se considerar a modelagem de problemas de aprendizado de máquina baseados no processamento de linguagem natural, as palavras remanescentes são as características do conjunto de dados sobre o qual deseja-se fazer o aprendizado.

2.3 Redução Dimensional

Ao utilizar base de dados extensas, ainda mais sendo composta por textos de domínios de conhecimento heterogêneos, é inevitável lidar com vetores de características extremamente longos. Além da elevação da complexidade computacional, o uso de representações vetoriais demasiadamente grandes pode não ser a opção mais adequada. Essa hipótese é confirmada no problema conhecido como "maldição da dimensionalidade", o qual expressa a existência um número ótimo de características que podem ser selecionados em relação ao tamanho da amostra para maximizar o desempenho do aprendizado [20]. Neste cenário, torna-se conveniente a aplicação de algum procedimento para redução da base de dados, seja pela seleção de características originais ou através de técnicas de redução da dimensionalidade. Esta última alternativa tem o objetivo de encontrar representações vetoriais menos complexas, criando novas características sintéticas a partir das originais.

Direcionada especialmente para representações vetoriais derivadas de textos, a **Indexação Semântica Latente** [2] (*Latent Semantic Indexing*, LSI) é uma técnica de redução dimensional baseada na Decomposição em Valores Singulares (*Singular Value Decomposition*, SVD). Sua adaptabilidade a dados de origem textual está atrelada a natureza esparsa dos dados. A LSI propõe construir um espaço "semântico" em que termos e documentos intimamente associados são colocados próximos uns dos outros.

[2]Também referenciada como Análise Semântica Latente (*Latent Semantic Analysis*, LSA) para propósitos para além da área de recuperação da informação.

Logo, supondo A como a matriz original $n \times m$, em que termos e documentos são representados em linhas e colunas respectivamente, a aplicação da LSI inicia-se pela adoção de um nível de aproximação k. Com isso, A pode ser decomposta da seguinte forma:

$$A \approx A_k = U_k D_k V_k^T \tag{2.4}$$

em que A_k é uma aproximação de A, composta pelo produto da matriz de termo-conceito U_k, a matriz de valores singulares D_k e a matriz de conceito-documento V_k. Assim, esta matriz A_k expressa a melhor representação da estrutura semântica do *corpus* original, omitindo todos, exceto os k maiores valores singulares na decomposição. Por tal razão, a LSI é também conhecida como SVD truncada [21, 22]. A respeito da escolha de k, esta é feita através de testes empíricos, avaliando a taxa de variância dos valores singulares. O valor de k deve ser pequeno o suficiente para permitir uma recuperação rápida da informação e grande o suficiente para capturar adequadamente a estrutura do *corpus*.

2.4 Métricas de Similaridade e Dissimilaridade

Medidas de similaridade e dissimilaridade desempenham um papel crítico na quantificação da semelhança ou distância semântica, respectivamente, entre textos. Independente dos elementos textuais comparados, caracteres, termos, *strings* ou *corpus*, tais medidas estão constantemente presentes na resolução de problemas de análise de padrões, sejam para fins de sumarização, classificação ou agrupamento de textos. Supondo um um par de vetores A e B não nulos, compostos pela mesma quantidade n de termos, tal que $A = [x_1, x_2, ..., x_n]$ e $B = [y_1, y_2, ..., y_n]$, é possível medir a relação semântica entre eles de diversas formas, tais como Distância Euclidiana, Distância de Manhattan e Similaridade do Cosseno.

A métrica de dissimilaridade conhecido como **Distância de Minkowski** é dada pela Equação 2.5. Tal métrica é uma generalização de outras duas igualmente conhecidas, a **Distância de Manhattan** e a **Distância Euclidiana**, para p igual a 1 ou 2 respectivamente. Obviamente, espera-se que quanto mais próximo de zero for o valor de Dis, mais similar A e B serão.

$$Dis(A, B) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{2.5}$$

Dentre as métricas de similaridade entre conjunto de termos, destaca-se a **Similaridade do Cosseno** que emprega o conceito de produto interno. Sendo definida entre $[-1, 1]$, valores dessa medida mais próximos ao limite superior simbolizam uma maior

proximidade entre os vetores de termos. Matematicamente, a similaridade do cosseno entre A e B é demonstrada pela equação:

$$Sim(A, B) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i-1}^{n} x_i^2}\sqrt{\sum_{i-1}^{n} y_i^2}}. \tag{2.6}$$

2.5 Aprendizado de Máquina

Após serem traduzidos para sua representação vetorial, os textos estão aptos para serem submetidos às técnicas de classificação, agrupamento, associação ou predição desempenhadas por algoritmos de aprendizado de máquina. Segundo Mitchell *et al.*, aprendizado de máquina é inerentemente um campo multidisciplinar, que trata a questão como construir programas de computador que melhoram automaticamente com a experiência [23]. Embora variações da definição de aprendizado de máquina coexistam, há um consenso na ideia de usar algoritmos para obter dados, aprender com eles e então determinar ou predizer algum fenômeno.

Existem diferentes algoritmos de aprendizado de máquina, cada qual indicado para um tipo de saída desejada. O aprendizado **supervisionado**, também intitulado como aprendizagem com exemplos, pressupõe a existência de entradas e saídas marcadas, compondo um conjunto de treinamento, para assim aprender uma regra geral que mapeia as entradas em saídas do modelo. Em contraste, o aprendizado **não supervisionado** não detém qualquer marcação sobre os dados, forçando o algoritmo identificar padrões entre as entradas, de modo que as entradas que têm algo em comum sejam agrupadas na mesma categoria. Já o **aprendizado por reforço** aprende a medida que interage com um ambiente dinâmico, e dessa maneira qualquer ação que tenha algum impacto no ambiente fornece um *feedback* que orienta o algoritmo.

2.5.1 Algoritmos supervisionados

A distinção dos algoritmos supervisionados pode ser feita definindo aqueles cujo resultado esperado são variáveis de valor real, intitulados algoritmos de regressão, e aqueles cujo resultado são categorias representadas por valores discretos, conhecidos como algoritmos de classificação. Devido à natureza classificatória das aplicações de processamento de linguagem natural propostas neste trabalho, os algoritmos de classificação são o foco.

2.5.1.1 Máquina de Vetor de Suporte

A Máquina de Vetor de Suporte, *Support Vector Machine* (SVM), consiste em um tipo de algoritmo classificador linear, baseado no conceito de um plano de decisão que define os limites de decisão. O processo decisório acontece através da geração de um hiperplano multidimensional ótimo que separa as amostras em classes, isto é maximizando a distância entre as classes ou a margem de separação. Tal hiperplano é traçado por um subconjunto de amostras, denominados vetores suporte. O caráter ótimo da separação é assegurado pela definição de uma função *kernel* que minimiza a função erro. Embora seja essencialmente um classificador binário, a SVM é igualmente adaptável a um problema multiclasse, em que divide-se o problema original em subproblemas de classificação binária.

Ao lidar com conjunto de amostras não linear, uma estratégia é adotar o artifício de utilizar uma função de *kernel*, em que uma função encontra um novo espaço dimensional, obrigatoriamente maior que o original, que viabilize a separação usando o hiperplano. Dentre as funções *kernels* mais utilizados estão a Linear, Polinomial, *Radial Basis Function* (RBF) e Sigmoid. A capacidade da SVM ser menos propensa ao sobreajuste (*overfitting*), ou seja, obter uma função de separação de complexidade superior à necessária, está intimamente relacionada ao grau de relevância atribuído amostras longe do limite de separação. Basicamente, uma vez encontrado o hiperplano, a maioria dos dados que não sejam os vetores de suporte são vistos como redundantes.

2.5.1.2 Floresta Aleatória

A Floresta Aleatória (*Random Forest*, RF) é um algoritmo popular de classificação ou regressão, que opera construindo múltiplas árvores de decisão durante o processo de treinamento. Durante o treinamento, a RF possibiliza a aplicação do método de *bagging*, que permite treinar repetidamente o algoritmo com o mesmo conjunto de dados entretanto selecionando as características aleatoriamente. Ilustrativamente, para um conjunto de treinamento para um conjunto de treinamento com $X = x_1, x_2, ..., x_n$ amostras de entrada e respectivos $Y = y_1, y_2, ..., y_n$ amostras de saída o método *bagging* implica a seleção aleatória e com repetição dessa base de dados K vezes. Assim, as árvores são treinadas com a mesma informação, de maneira que o resultado final é formado pelas predições individuais m_i de cada árvore do conjunto conjunto, conforme a equação:

$$\hat{m} = \frac{1}{K} \sum_{k}^{i=1} m_i \tag{2.7}$$

Um vantagem relevante da RF para ao modelo tradicional de árvores de decisão é o fato de não ser considerado todo o conjunto de dados, mas apenas um subconjunto dele. Isto implica uma maior aleatoriedade no modelo, auxiliando na correção do sobreajuste. No mesmo sentido, ao incrementar o número de árvores de decisão na RF, a taxa de erro do conjunto de testes converge para um limite, significando que RF mais povoadas são menos suscetíveis ao sobreajuste [24].

2.5.1.3 k-Vizinhos Mais Próximos

O algoritmo k-Vizinhos Mais Próximos (*k-Nearest Neighbors*, kNN) depende da escolha prévia de um parâmetro k, que condiciona o número de amostras vizinhas mais próximas usado no critério de classificação. A partir de uma amostra ainda não classificada, o algoritmo aplica uma métrica de distância entre essa amostra e todas as demais já classificadas. Filtrando as k amostras vizinhas que tiveram as menores distâncias, o algoritmo verifica e contabiliza a quantidade de amostras integrantes em cada classe. Finalmente, a amostra é alocada na classe majoritária dos k vizinhos mais próximos. Essa dependência sobre o valor do parâmetro inicial faz com que o resultado do algoritmo apresente diversas classificações, se k for muito alto, ou apresente amostras ruidosas, se k for muito pequeno. Ao ser obrigado a calcular a distância de cada amostra nova com todas as demais já classificadas, o algoritmo requer um consumo computacional maior, sendo assim não indicado para *corpus* muito grandes [25].

2.5.2 Algoritmos não supervisionados

Algoritmos de agrupamento são a forma mais comum de aprendizado não supervisionado. Apesar de possuírem lógica operacional, caso de uso, escalabilidade e desempenhos distintos, o propósito genérico de usar esses algoritmos é a segregação de termos em grupos (*clusters*) de acordo com suas características semânticas. Esse procedimento de separação em grupos é conhecido como clusterização.

2.5.2.1 Algoritmos Baseados no Particionamento

Essa classificação é dada àqueles algoritmos que são semelhantes no sentido de que cumprem simultaneamente dois critérios no processo de agrupamento de dados. O primeiro critério expressa a obrigação de ter pelo menos uma amostra em cada agrupamento criado. O segundo refere-se a uma exclusividade de pertencimento, em que cada amostra

deve pertencer a somente um agrupamento [26, 27].

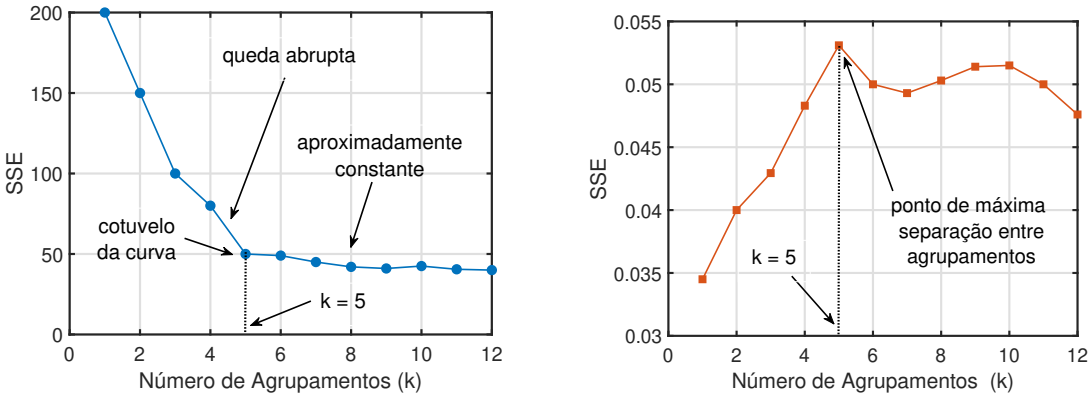
Um exemplar clássico desse tipo de algoritmo é o ***k-means***, uma heurística capaz de particionar dados em k agrupamentos pela minimização da soma dos quadrados das distâncias em cada agrupamento. Sua lógica de execução, parte da escolha aleatória dos centróides de cada agrupamento seguida do cálculo de distância entre cada amostra e os centróides, segundo uma das métricas de dissimilaridade, ou similaridade, vistas na Seção 2.4. Posteriormente cada amostra é alocada no agrupamento cujo centroide está mais próximo. A cada nova amostra alocada a um agrupamento, o centróide é recalculado podendo ocorrer eventuais redistribuições de amostras para outros grupos. O algoritmo finaliza quando cessam essas alterações na alocação das amostras aos agrupamentos.

Outro exemplo é o algoritmo ***k-medoids***, indicado para pequenos conjuntos de dados, e que também particiona os dados em k grupos adotando o critério de minimizar a soma dos quadrados das distâncias em cada grupo. Embora lembre o *k-means*, sua inovação está no fato de escolher efetivamente uma das amostras de entrada como centro dos agrupamentos, não pontos médios igual ao *k-means*. Essa característica de tomada de decisão se traduz em maior robustez a dados ruidosos e *outliers*, além de uma capacidade de lidar com alta dimensionalidade, útil em representações vetoriais de dados textuais [26, 27].

Contudo ambos os algoritmos, assim como outros, estão sujeitos à uma desvantagem singular: a indeterminação quanto ao número adequado de grupos k. A fim de contornar essa indeterminação, são usados dois métodos, *Elbow* e da *Silhueta*, para analisar previamente a conformidade dos dados a quantidades diferentes de grupos e, assim, obter um resultado adequado aos dados. Em particular, o *Elbow* mede a compactação dos agrupamentos estabelecendo uma relação entre o número de agrupamentos e sua influência na variação total dos dados dentro do grupo. Graficamente, o melhor valor de k é encontrado identificando o ponto em que o ganho da curva diminui drasticamente, permanecendo aproximadamente constante depois disso. Da mesma análoga, o método da Silhueta mede a qualidade de um agrupamento. O número ideal de agrupamentos k é aquele que maximiza a silhueta média em uma faixa de valores possíveis para k [28, 29].

2.5.2.2 Algoritmos Baseados em Densidade

Algoritmos de agrupamento baseados em densidade compartilham uma relação próxima com a abordagem do vizinho mais próximo (*nearest neighbour*). Nesse sentido, um agrupamento, definido como um componente denso conectado, cresce em qualquer direção



(a) Método *Elbow* (b) Método da Silhueta

Figura 2.2: Métodos complementares para determinar o número ótimo de agrupamentos. Ambos os métodos idealmente tendem a convergir para um mesmo k, verificado neste exemplo como k=5.

que a densidade conduza. Esta lógica de formação dos agrupamentos está diretamente relacionada à principal vantagem desses algoritmos em relação ao grupo dos algoritmos de particionamento, a possibilidade de descobrir agrupamentos com formas arbitrárias, diferente dos tipicamente agrupamentos esféricos retornados pelo algoritmo *k-means*, por exemplo.

Dentre os algoritmos baseados em densidade, o **DBSCAN** (Clusterização Espacial Baseada em Densidade de Aplicações com Ruído, *Density Based Spatial Clustering of Application with Noise*) é o mais popular. Seu intuito é encontrar regiões que satisfaçam uma densidade de pontos mínima estabelecida e que sejam separadas por regiões de menor densidade. Para isso, o algoritmo realiza uma estimativa simples do nível de densidade mínimo, definindo um limite para o número de vizinhos, $minPts$, dentro de um raio ϵ. Assim, uma amostra com mais de $minPts$ vizinhos dentro desse raio, é considerada um ponto central. Analogamente, uma amostra é considerada como de borda, se dentro de sua vizinhança concentram-se menos amostras que o mínimo definido, porém ainda pertencem à vizinhança de um ponto central qualquer. Por último, amostras que não são alcançáveis por densidade a partir de qualquer ponto central, ou seja, não se configuram nem como pontos centrais nem de borda, são rotulados como *outliers*. Uma desvantagem associada ao seu uso consiste na sua complexidade fortemente polinomial, que requer $\Omega(n^{\frac{4}{3}})$ tempo para convergir, em que n é o tamanho do conjunto de dados [27, 30, 31].

2.5.2.3 Algoritmos Hierárquicos

Os hierárquicos não apenas criam agrupamentos, mas consideram uma lógica multinível e calculam uma representação hierárquica dos dados de entrada. Esta representação é um tipo particular de árvore, em que os nós-folha expressam dados individuais, e pode ser construída seguindo um método aglomerativo ou divisivo. O método aglomerativo, conhecido também como abordagem *bottom-up*, começa considerando cada amostra como um único agrupamento e mescla recursivamente duas ou mais em um novo agrupamento seguindo uma métrica de similaridade. Por outro lado, o método divisivo, e.g. abordagem *top-down*, inicia com uma estrutura plana em que todas as amostras pertencem ao mesmo agrupamento, ou seja, nível hierárquico. Portanto, a cada iteração, o algoritmo divide um ramo-pai em dois subconjuntos menores, os ramos-filhos. O processo termina quando um critério de parada é atingido, frequentemente, o número k de agrupamentos. No final do algoritmo, é criado um dendrograma de agrupamentos, uma hierarquia de árvore binária similar [32, 13, 27].

2.5.3 Métricas de Avaliação

Independente do algoritmo, supervisionado ou não supervisionado, caso haja o conhecimento prévio sobre dados rotulados com base em uma verdade básica (*ground truth*), torna-se plausível a clara identificação de quantidade de Verdadeiros Positivos (VP), Falsos Positivos (FP), Verdadeiros Negativos (VN) e Falsos Negativos (FN). Tais classificações compõe o cálculo de várias métricas de recuperação de informação, como:

- **Acurácia** (A_c) é definida pela razão do total de amostras classificadas corretamente (VP + VN), pelo número total de amostras (P+N). Para conjunto de dados não-balanceados, uma avaliação de desempenho baseada exclusivamente nesta métrica pode gerar conclusões erradas;

- **Precisão** (P_r) é a razão entre, dada uma classe alvo, a quantidade de amostras corretamente classificadas para a classe em questão (VP), pelo conjunto total de predições atribuídas a essa classe, isto é, corretas e incorretas (VP + FP);

- **Sensibilidade** (S_s) também conhecida como revocação (*recall*) ou **taxa de verdadeiros positivos** é definida pela razão entre a quantidade de amostras corretamente preditas (VP) para um classe positiva e o total de amostras que pertencem a esta classe, incluindo assim tanto predições corretas quanto as que deveriam ter indicado

esta classe (VP + FN). O análogo para a classe negativa é chamado de **especificidade** ou **taxa de verdadeiros negativos**;

- **Medida-F_1** (F_1-*Score*) relaciona a precisão e a sensibilidade por uma média harmônica expressa por

$$Medida - F_1 = \frac{2}{\frac{1}{P_r} + \frac{1}{S_s}}; \tag{2.8}$$

  Geralmente, quando maior o valor da medida-F_1, melhor a classificação sendo um reflexo do compromisso mútuo entre a precisão (P_r) a sensibilidade (S_s):

- **Área abaixo da curva ROC** é medida através da curva Característica de Operação do Receptor (ROC), uma representação da razão entre a taxa de verdadeiros positivos e a taxa de verdadeiros negativos, para vários limiares. Essa curva descreve graficamente o desempenho de um modelo de classificação. Sucintamente, quanto maior a área abaixo da curva, melhor o desempenho do modelo.

Quando a referência da correta classificação sobre o conjunto de dados é desconhecida, torna-se impossível qualquer avaliação dos resultados gerados por algoritmos de aprendizado não-supervisionado, seja com base em métricas de recuperação de informação, ou mesmo com base em medidas extrínsecas, como informações mútuas, homogeneidade ou completude. Dessa forma, cabe uma avaliação baseada em medidas intrínsecas, que quantificam a coesão e separação dos agrupamentos [33]. Dentre essas medidas pode-se citar:

- **Davies-Bouldin**, um índice que fornece uma estimativa do grau de sobreposição do agrupamento. É definido como a medida de similaridade média de cada agrupamento com seu par mais semelhante, em que a similaridade é a razão entre as distâncias dentro do agrupamento e entre as distâncias entre os agrupamentos. Ao considerar o pior cenário de similaridade para cada agrupamento, espera-se que quanto mais próximo de zero, o valor mínimo, melhor será o índice e, consequentemente, melhores serão os resultados do processo de agrupamento [34, 35];

- **Calinski e Harabasz**, uma razão entre a dispersão dentro do agrupamento e a dispersão entre os agrupamentos. Ao contrário do índice anterior, seu valor ótimo maximiza a pontuação [36];

- **Coeficiente da Silhueta**, um índice definido entre $[-1, +1]$, em que quanto mais próximo a 1, melhor a adequação das amostras aos agrupamentos aos quais perten-

cem. De maneira oposta, valores próximos a 0 ou negativos informam que o processo de agrupamento tem muitos ou poucos agrupamentos.

Capítulo 3

Trabalhos Relacionados

Banko *et al.* destacam a defasagem entre a quantidade de textos disponível online, da ordem de centenas de bilhões de palavras, e a quantidade que é efetivamente usada como *corpora* de treinamento no aprendizado de máquina aplicados ao PLN. A fim de compreender este cenário, os autores conduzem uma avaliação do desempenho de diferentes métodos de aprendizado, incluindo Winnow, Percéptron, Naïve Bayes, em uma tarefa de desambiguação de linguagem natural, quando treinados em ordens de grandeza de até 1 bilhão de dados rotulados [37].

Em contrapartida, alguns trabalhos concentram-se em apresentar novas soluções de ferramentas para o processamento de linguagem natural. Manning *et al.* apresentam o *Starford CoreNLP*, um arcabouço leve, desenvolvido em `Java`, focado na análise e interpretação linguística de dados em linguagem natural. Diferentemente de outros softwares preocupados com documentação e escalabilidade de máquinas, sua arquitetura é exclusivamente dedicada a gerar textos anotados[1] a partir de textos inseridos sem qualquer tratamento. Tal procedimento é realizado através da análise em diferentes níveis linguísticos seguindo uma ordem editável pelo usuário. Apesar disso, seu potencial só é totalmente aproveitado com entradas na língua inglesa [18]. Outros trabalhos, como o de Socher *et al.*, propõem um par de ferramentas complementares para prever com precisão os efeitos semânticos composicionais presentes em um conjunto de textos. A primeira delas, o *Stanford Sentiment Treebank*, consiste em uma base de dados contendo sentenças em inglês marcadas quanto a sua polaridade. A segunda ferramenta, o *Recursive Neural Tensor Network*, é um modelo que aplica um algoritmo recursivo que permite uma representação vetorial composicional de frases com tamanho e tipo sintático variável. Comparações do

[1] Textos anotados são textos enriquecidos com informações adicionais usadas para explicar características linguísticas ou textuais.

modelo proposto com outros, Rede Neural Recursiva, *Matrix-Vector RNN*, e classificadores conhecidos como o Naïve Bayes, Bigramas e SVM, avaliam a acurácia da obtenção de sentimento em várias formas: utilizando agrupamento das N palavras mais frequentes, em frases com cunho negativo negadas e com cunho positivo negadas [38].

Estudos como o de Runeson *et al.* incorporam técnicas de análise linguística computacional na identificação de duplicatas em relatórios de defeitos em softwares, comumente geradas devido ao paralelismo no desenvolvimento. A precisão e a sensibilidade são as métricas usadas na etapa de avaliação da ferramenta. Auxiliado por essas métricas foram feitos testes de similaridade nos relatórios, variando o uso das técnicas de PLN [39].

Percebe-se que a aplicabilidade das técnicas de processamento de linguagem natural é bastante ampla. Restringido o escopo de atuação do PLN para as áreas de interesse deste trabalho, a sumarização de textos, a detecção de notícias falsas e a construção de ontologias, a seguir é apresentada uma revisão sobre o estado da arte desses assuntos.

3.1 Sumarização Automática de Textos

Cardoso *et al.* desenvolveram métodos de sumarização automática de textos baseados em dois modelos de discurso semântico, Teoria da Estrutura Retórica (RST) e Teoria da Estrutura entre Documentos (CST). Enquanto o modelo RST detalha os principais aspectos da organização de um documento textual e indica unidades relevantes do discurso, o modelo CST descreve conexões semânticas entre unidades de textos relacionados. Os autores utilizam o CSTNews [40], um *corpus* dedicado ao processamento de documentos múltiplos, composto por 50 grupos de artigos escritos em português, coletados de diferentes seções das principais agências de notícias. Cada grupo possui aproximadamente três artigos jornalísticos de fontes diferentes, abordando o mesmo assunto. Cada grupo de notícias tem cinco resumos extrativos. Os resultados mostram que o uso do conhecimento do discurso semântico melhora a informatividade dos resumos automáticos [41].

Oliveira *et al.* promoveram uma extensa análise comparativa de dezoito técnicas de ponderamento textual para calcular a importância de uma frase no contexto de resumos extrativos de documentos únicos e múltiplos. Dezoito medidas de ponderação são extraídas do sítio *web* da CNN[2] e dos conjuntos de dados de referência DUC[3], como similaridade com o título, *TextRank* e centralidade, que subsequentemente alimentam uma série de al-

[2]Disponível em: https://edition.cnn.com/
[3]Disponível em https://www-nlpir.nist.gov/projects/duc/index.html

goritmos de aprendizado de máquina. A avaliação é feita em muitos cenários de aplicação: individualmente, combinando e comparando-os com os resumos de última geração usando medidas de avaliação tradicionais, como a sensibilidade da *Rouge-1* e a sensibilidade da *Rouge-2* [42].

3.2 Criação de Estruturas Ontológicas

Recentemente, vários estudos têm se dedicado a buscar uma abordagem automática para formar ontologias a partir de textos. Integrando esta busca, Rani *et al.* apresentam uma abordagem semi-automática dupla para construir uma ontologia de termos mais rica e um grafo ontológico com intervenção humana mínima. Usando o algoritmos de modelagem de tópicos, LSI & SVD e *MapReduce Latent Dirichlet Allocation* (Mr.LDA), seu estudo classifica tópicos e palavras associadas para gerenciamento de conhecimento e recuperação semântica [43].

Tendendo ao uso de estruturas sintáticas na codificação do conhecimento humano, Petrucci *et al.* propõem um sistema que explora redes neurais recorrentes na configuração codificador-decodificador, para traduzir as definições da linguagem natural em uma fórmula lógica. O modelo mostra-se capaz de tolerar palavras desconhecidas e melhorar seu desempenho ao enriquecer o conjunto de treinamento com a adição de novas frases anotadas [44].

Com foco nas aplicações práticas da ontologia, Kaiya *et al.* desenvolvem um método que visa apoiar analistas de requisitos no processo de elicitação de requisitos, considerando o aspecto semântico de descrições escritas em linguagem natural. Por meio de uma abordagem experimental, em um estudo de caso de software de reprodução de música, a facilidade de uso do método e a eficácia foram avaliadas [45].

Outros trabalhos dedicam-se ao estudo de uma versão particular da ontologia, a taxonomia. Babbar *et al.*, por exemplo, investigam os impactos de uma grande quantidade de dados na construção de taxonomias com classificadores planos e hierárquicos. Seu trabalho também propõe a definição de limites que auxiliam na identificação do erro de classificadores implantados em taxonomias de grande escala e uma técnica baseada em poda que melhora o procedimento de classificação taxonômica [46]. Um propósito alternativo para implementar a taxonomia são encontrados em Liu *et al.* [47]. Em seu estudo, a taxonomia é usada em uma estrutura de interpretação *post-hoc* para entender as incorporações de rede. Os autores descrevem a extração da estrutura básica da taxonomia

através da aplicação de um algoritmo de agrupamento hierárquico sobre incorporações de *node2vec*, um algoritmo de vetorização. Só então as características globais são capturadas. Da mesma forma, Zouaq *et al.* realizam uma extensa investigação com seis incorporações de grafo de conhecimento, associados a estatísticas de dados vinculados e métodos de agrupamento, para avaliar sua capacidade de extrair uma taxonomia expressiva [48].

Diferentemente, Woon *et al.* apresentam uma abordagem para organizar automaticamente palavras-chaves em uma estrutura hierárquica, buscando refletir com precisão as inter-relações entre os termos da área de energias renováveis. Na metodologia usada, emprega-se a frequência da co-ocorrência de termos como um indicador da proximidade semântica entre os termos [49].

3.3 Detecção de Notícias Falsas

A identificação de conteúdos maliciosos ou falsos e seus disseminadores apresenta-se como um objeto de pesquisa frequente na literatura. Sob a hipótese de que as notícias falsas tentam tornar as histórias interessantes aos leitores, Rashkin *et al.* apresenta um estudo analítico para caracterizar a linguagem de citações políticas e mídia de notícias de intenções diversas e diversos graus de verdade [50]. O método empregado conta com o uso do LIWC (Consulta Linguística e Contagem de Palavras, *Linguistic Inquiry and Word Count*) [51], um software de análise textual que revelou características léxicas latentes. A análise dessas características permitiu formular perfis distintos de notícias dependendo da sua fonte de veiculação. Assim, constata-se que notícias oriundas de fontes confiáveis normalmente apresentam alguma forma de embasamento concreto, como comparações numéricas e expressões relativas a dinheiro. Em um sentindo oposto, notícias de fontes menos confiáveis detinham um incidência maior de pronomes de primeira e segunda pessoa, superlativos, advérbios de modo e palavras que expressão hesitação. Outras abordagens concentram-se em um tipo específico de notícias falsas. Especialmente, Rubin *et al.* buscam comparar notícias de sátira com seus pares legítimos, usando a SVM enriquecida com cinco características preditivas, Absurdo, Humor, Gramática, Afeto Negativo e Pontuação [52]. Chen *et al.* propõem uma discussão preliminar do *clickbait*[4], caça-cliques, como um exemplo de notícias falsas ou enganosas [53]. Seu trabalho analisa as características de identificação e apresenta potenciais métodos para a detecção dessa fraude.

Algumas propostas, além de apresentar métodos de identificação, visam também com-

[4]Tipo de propaganda falsa que usa imagens em miniaturas, ou manchetes sensacionalistas, para atrair a atenção e estimular usuários a acessarem e compartilharem o conteúdo.

por bases de dados de notícias, que na sua maioria em Inglês, tais como o Emergent [54] e o LIAR [55]. Em Português, é possível encontrar alguns sítios *web* que compilam notícias verdadeiras e falsas para a verificação da veracidade. Contudo, esses sítios *web* normalmente apresentam apenas os comentários referentes a essas notícias e não as versões originais. Contrariando essa realidade, Monteiro *et al.* disponibilizam o Fake.Br Corpus, um *corpus* composto por notícias falsas e legítimas que foram manualmente selecionadas [56].

Outros trabalhos, como Barreto *et al.*, propõem uma metodologia capaz de distinguir usuários legítimos e *spammers* considerando a 2-vizinhança no *Twitter*. A proposta é subdivida em três etapas, cuja primeira é a pré-seleção manual de possíveis usuários. Como critério de pré-seleção de um usuário malicioso utiliza-se o fato do usuário enviar mensagens contendo pelo menos um tópico popular. A segunda etapa inclui a coleta dos dados da rede no entorno dos usuários pré-selecionados. Como última etapa, é feita uma análise desses dados avaliando métricas como distribuição de grau, centralidade de grau, coeficiente de agrupamento e *PageRank*. Ao final, os autores relatam um comportamento diferenciado da distribuição de grau dos *spammers*, contrariando a lei de potência esperada para os usuários legítimos [57].

Capítulo 4

Sumarização Automática de Textos

Um resumo gerado automaticamente deve conter as informações mais relevantes de um documento, cobrindo o máximo número de tópicos, ao mesmo tempo em que ocupa o mínimo de espaço. No entanto, a geração automática de resumos é desafiadora, devido à existência de diversos problemas, como redundância, dimensão temporal, referências cruzadas, ordenação de sentenças, entre outros, que requerem atenção especial ao resumir um documento [58].

Os métodos automatizados para resumos de texto são classificados em resumo abstrativo ou extrativo [3]. Os resumos abstrativos usam métodos linguísticos para examinar e interpretar o texto e, em seguida, encontrar os novos conceitos e expressões para melhor descrevê-lo, gerando um novo texto mais curto que transmita as informações mais importantes do documento de texto original. Os resumos extrativos são formulados extraindo segmentos mais relevantes do texto, com base na análise estatística de características individuais ou combinadas, como frequência de palavras ou frases, localização ou palavras-chave para localizar as frases a serem extraídas.

Embora sejam utilizadas técnicas diferentes para a geração de resumos abstrativos e extrativos, ambos constituem problemas complexos e com múltiplas variáveis. Ao gerar resumos abstrativos, o principal desafio é o da representação semântica. A capacidade de geração de resumos é limitada pela riqueza da representação e pela capacidade de gerar estruturas de representação, já que as ferramentas não são capazes de resumir o que suas representações semânticas não podem capturar [3]. É possível criar estruturas de representação apropriadas em um domínio limitado, por exemplos ontologias usadas no meio médico [59], mas uma solução de propósito geral depende da análise semântica em um domínio aberto. No resumo extrativo, o principal desafio é evitar prejuízo à coerência do texto gerado. As frases extraídas geralmente são mais longas que a média da frases

no texto, pois estatisticamente carregam mais informação. Assim, os resumos tendem a conter partes de frases que não são essenciais para o resumo, mas que são incluídas, porque a unidade sobre a qual a decisão ocorre é a sentença completa. A extração leva a problemas na coerência geral do resumo, porque eventualmente as frases extraídas fazem referência a termos que estão em frases não selecionadas, como pronomes, que perdem suas referências quando extraídos do contexto. A junção de extratos descontextualizados pode ainda levar a interpretações enganosas ao se considerar referências anteriores falsas. Esses desafios são contornados por pós-processamento dos fragmentos extraídos, substituindo pronomes por seus antecedentes, substituindo expressão temporal relativa por referências absolutas, entre outras medidas.

A geração automática de resumo apoiada por aprendizado de máquina pode ser supervisionada ou não supervisionada [60]. Os dados de treinamento são necessários na geração supervisionada para selecionar conteúdo importante dos documentos, sendo essencial uma grande quantidade de dados rotulados ou anotados para treinar as técnicas. Sistemas de resumo automático com aprendizado supervisionado consideram que cada frase é uma amostra de um problema de classificação de duas classes, no qual as frases pertencentes ao resumo são discriminadas como amostras positivas e as demais como negativas. Os sistemas que aplicam métodos de aprendizado não supervisionado analisam apenas os documentos fontes do resumo.

Estes sistemas aplicam regras heurísticas para extrair sentenças altamente relevantes e gerar um resumo. Em geral, os algoritmos empregados em sistemas não supervisionados são algoritmos de agrupamento, tais como *k-means* e *k-medoids* [60]. Vale ainda ressaltar que os métodos de resumo podem ser classificados de acordo com a fonte, que pode ser compilada em um documento único ou vários documentos. No resumo de documento único, apenas um documento é fornecido para geração de resumo. Os métodos extrativo e abstrativo podem ser aplicados no resumo de um único documento [2].

As principais abordagens [60] para gerar resumos automatizados consistem na seleção de frases do texto original baseando-se em abordagens que: (i) consideram a importância da palavra no texto, mensurada através do método de representação vetorial TF-IDF; (ii) se baseiam na Análise Semântica Latente (*Latent Semantic Analysis* - LSA); (iii) se baseiam em mecanismo de aprendizado de máquina como algoritmos classificação, agrupamento, controladores de lógica nebulosa [61], redes neurais [62] e, até mesmo, mecanismos de regressão; (iv) utilizam abordagens baseadas em métricas de centralidade, similaridade e grau de nós através da representação do texto em grafos; ou (v) adotam a

extração de conceitos baseada em redes de conhecimento e, então, usam os conhecimentos obtidos como características para o problema de otimização [63].

4.1 Rezzumin: A Ferramenta Proposta de Sumarização Automática

Esta seção apresenta a Rezzumin[1], uma ferramenta *web* que implementa uma abordagem híbrida de geração de resumos extrativos automatizados em português. Permitindo textos em diferentes formatos, a Rezzumin aplica diferentes técnicas de processamento de linguagem natural, incluindo uma normalização de palavras por meio de um tesauro, *i.e.*, um dicionário de sinônimos, com o intuito de mapear todos os sinônimos em uma única palavra. A ferramenta ainda aplica uma heurística gulosa sobre um grafo de conhecimento, para expandir o número de tópicos cobertos no resumo e um mecanismo de aprendizado não supervisionado para selecionar apenas as sentenças mais relevantes para o resumo final.

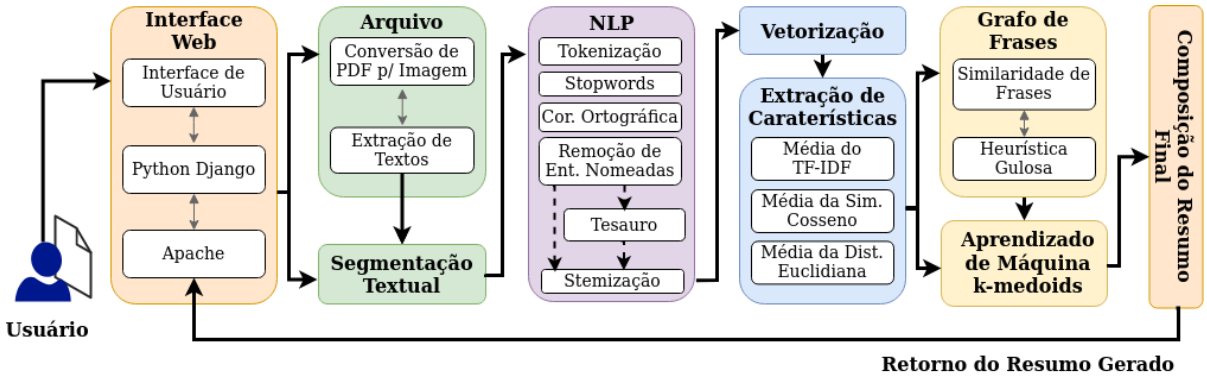


Figura 4.1: A arquitetura da ferramenta Rezzumin é composta por vários módulos. O processamento do texto ocorre no em um servidor *web* e, após a geração do resumo, o resultado é devolvido ao usuário.

A arquitetura da ferramenta Rezzumin é modular e coesa, mostrada na Figura 4.1, permitindo a adequação de cada módulo para cada idioma específico. Na versão atual, a ferramenta foca no português. O módulo de **Interface *Web*** é implementado sobre um servidor Apache 2.2, executando o módulo Django para suportar aplicações `Python`. O módulo de **Arquivo** converte arquivos PDF submetidos pelo usuário em texto plano, através do `tesseract`[2], um aplicativo de reconhecimento óptico de caracteres (*Optical*

[1]A ferramenta é de código aberto e distribuída sob a licença *Creative Commons 4.0 International*. A ferramenta Rezzumin, o código fonte, o vídeo de apresentação e os manuais de uso estão disponíveis em: http://rezzumin.labgen.lid.uff.br/

[2]Disponível em: https://opensource.google/projects/tesseract.

Character Recognition - OCR). O módulo de **Segmentação Textual** inicia o processamento de texto, seja vindo do OCR, seja vindo diretamente através da interface *web*. Sua função é fragmentar o texto contíguo original em sentenças. O módulo de **Processamento de Linguagem Natural** (PNL) é implementado na linguagem `Python` com base na biblioteca `NLTK` [3] e é responsável pelo pré-processamento dos dados. O módulo executa as tarefas de limpeza e adequação dos dados textuais, para então alimentar o módulo de **Vetorização**. A vetorização leva as palavras do texto para um espaço vetorial cuja dimensionalidade é igual ao número de palavras restantes após os processos de limpeza e adequação. Com os dados vetorizados, o módulo de **Extração de Características** converte os vetores de palavras em características úteis, que são tratadas pelos módulos de **Grafo de Frases** e **Aprendizado de Máquina**. A vetorização e a extração de características utilizam funções providas pela biblioteca `Scikit-learn` [4] e a geração do grafo utiliza a biblioteca `NetworkX`[5]. Por fim, os resultados são reunidos no módulo de **Composição do Resumo Final** e o resultado é retornado para o módulo de interface *web*.

O principal diferencial da Rezzumin concentra-se no uso de resumos abstrativos como os resumos de referência no processo de avaliação, enquanto que trabalhos anteriores empregam resumos extrativos. Dessa forma, constata-se que a metodologia de avaliação utilizada nesses trabalhos é probabilisticamente menos abrangente, pois é assegurado *a priori* que, se o método de extração for eficiente, será possível produzir um resumo de saída literalmente idêntico ao resumo de referência. Esta garantia não é mantida se o resumo de referência for abstrativo.

4.2 Conversão e Processamento Textual

A Rezzumin permite o *upload* de textos em diferentes formatos, tanto no formato de texto simples, quanto em PDF. Para este último formato, a ferramenta ainda traz uma otimização no tratamento de artigos da Sociedade Brasileira de Computação (SBC). Em especial para documentos em PDF, a Rezzumin executa o *tesseract*, uma vez que a codificação de textos em arquivos PDF, por vezes, compromete a conversão do texto codificado em texto plano. Assim, a leitura do arquivo depende do reconhecimento óptico do texto. Em posse do texto plano, a ferramenta o submete às etapas 1-6a do processamento textual descrito na Seção 2.1, correspondendo as etapas de *tokenização*, remoção de pontuação e caracte-

[3]Disponível em: https://www.nltk.org/.
[4]Disponível em: https://scikit-learn.org/stable/.
[5]Disponível em: http://networkx.lanl.gov/.

res especiais, eliminação de *stopwords*, correção ortográfica, reconhecimento de entidades nomeadas e *stemização*, respectivamente. Como procedimento adicional entre as etapas 5 e 6a, é proposto um mapeamento de palavras utilizando um tesauro [6], um dicionário de sinônimos de aproximadamente 33 mil verbetes. Conforme descrito no Algoritmo 1, o objetivo é reduzir a pluralidade de palavras, sem promover uma perda semântica abrupta, trocando-as pelo sinônimo mais utilizado. O registro da frequência de ocorrência no documento de cada palavra do tesauro é atualizado constantemente cada vez que a ferramenta trata um documento. Após o processamento, cada sentença, então vista como um vetor de radicais, é convertida na sua representação vetorial segundo o modelo TF-IDF.

Algoritmo 1: Implementação do Tesauro

```
Entrada: Texto é uma lista de todas as sentenças do texto
         DicSin é o dicionário de sinônimos de palavras
         DicFreq é o dicionário contendo a frequência de cada sinônimo
         SinCandidato() é uma função que retorna o sinônimo mais frequente
for sentença ∈ Texto do
    NovaSentença ← [ ] for palavra ∈ sentença do
        if palavra ∈ DicSin.chaves() then
            sinônimo ← SinCandidato(DicSin[palavra]) DicFreq[palavra] ←
             DicFreq[palavra] + 1 palavra ← sinônimo
             NovaSentença.adiciona(palavra)
        else
            NovaSentença.adiciona(palavra)
    retorna NovaSentença
```

4.3 Extração de Características e Grafo de Frases

As características extraídas do texto podem ser divididas em duas origens, aquelas oriundas diretamente de métricas vetoriais e aquelas que tem como base o grafo de frases. A respeito das características de cunho vetorial, estas foram derivadas segundo 3 tipos de cálculos: a média do somatório do valor TF-IDF de cada frase; a média do somatório da similaridade do cosseno de cada frase em relação às demais; e a média do somatório da distância euclidiana de cada frase em relação às demais.

Para encontrar as demais características, cria-se um grafo bidirecional no qual um cada vértice representa um sentença do texto e cada aresta é ponderada pelo resultado da similaridade do cosseno entre as frases. Como critério mínimo de existência de uma aresta, é estabelecido que o resultado da similaridade deve ser maior ou igual a 0,1. Uma vez

[6] Disponível em: http://www.nilc.icmc.usp.br/tep2/ajuda.htm.

em posse do grafo bidirecional, pode-se extrair dois conjuntos de características relacionadas ao nível de conectividade de cada nó, a centralidade de intermediação (*betweenness centrality*) e a centralidade de grau (*degree centrality*). A partir desse grafo, executa-se também uma heurística de descoberta das frases mais centrais para o texto [64]. A ordenação das frases ocorre como se segue. A cada iteração a heurística seleciona a frase de maior centralidade no texto, a coloca em uma lista ordenada e a retira do grafo. Esse procedimento é repetido até que o grafo esteja totalmente desconexo. A interpretação de frases mais centrais é que essas são as com maior semelhança para a maioria das frases na sua vizinhança. Após a seleção do subconjunto de frases mais centrais, as amostras de frases vetorizadas são enriquecidas com variáveis binárias que indicam de qual frase do conjunto de frases centrais a amostra está mais próxima.

Sobre esse conjunto de características formado, executa-se o algoritmo de *k-medoids* para a seleção das frases que melhor representam o texto original. O valor de k é um parâmetro fornecido pelo usuário que determina a fração de frases que deve ser contida no resumo gerado. Vale ressaltar que todas as variáveis usadas no algoritmo são normalizadas para evitar a predominância de algumas variáveis sobre as demais.

4.4 A Avaliação da Ferramenta Rezzumin

A avaliação da proposta compara o uso de características extraídas apenas da vetorização do texto (`Espaço Vetorial`), apenas do grafo de relação entre sentenças (`Grafo`) e todo o conjunto de características disponíveis (`Híbrida`). A primeira métrica de avaliação é a *Rouge-1*, que traduz a qualidade de um resumo gerado ao compará-lo com um resumo de referência, derivando três medidas de recuperação de informação: precisão, sensibilidade e medida-F1. Os resultados mostrados nas Figuras 4.2 e 4.3 são médias com intervalo de confiança de 95% sobre resumos executados em artigos em português.

Com base na Figura 4.2 é notória a homogeneidade entre as medidas de precisão, sensibilidade e medida-F1 das metodologias aplicadas. Contudo, a metodologia híbrida enriquecida com tesauro apresenta um comportamento consistente em todas as métricas e melhora o desempenho das abordagens clássicas. Ciente de que os conjuntos de resumos de referência utilizados na avaliação foram retirados de artigos científicos publicados em edições anteriores do Simpósio Brasileiro de Redes de Computadores (SBRC), sendo portando escritos por autores humanos, assume-se que estes sejam coesos, sucintos e informativos. Diante dessa perspectiva, resumos extrativos que englobem 1/3 da quantidade

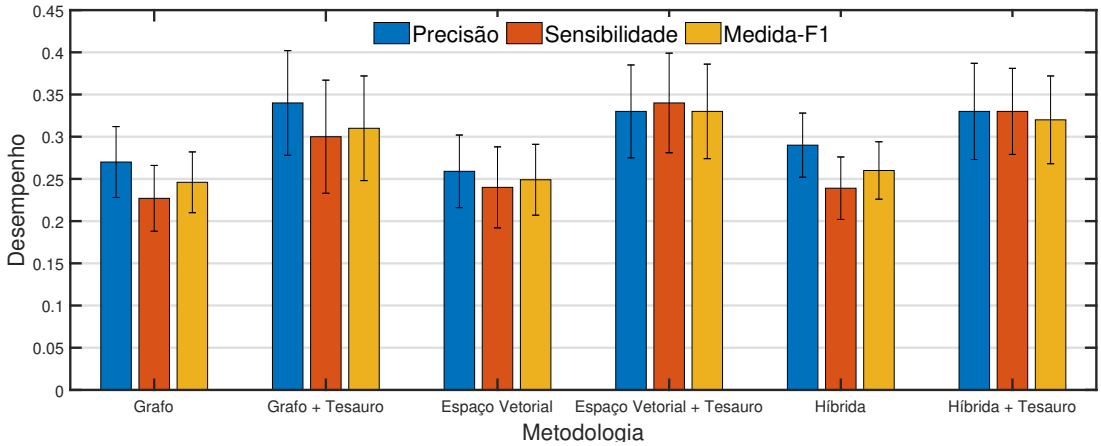


Figura 4.2: Avaliação dos métodos usados para gerar resumos com o Rezzumin. A abordagem `Híbrida + Tesauro` supera em até 28% as abordagens clássicas de `Espaço Vetorial` e `Grafo`.

de informação condensada, como os gerados pela ferramenta Rezzumin, são considerados resumos informativos. Também é importante destacar um aumento significativo de até 28% no desempenho de todas as metodologias que foram enriquecidas com o tesauro. Esse crescimento se deve à capacidade do tesauro de encontrar uma intersecção semântica entre palavras com radicais diferentes, substituindo-as por um termo comum.

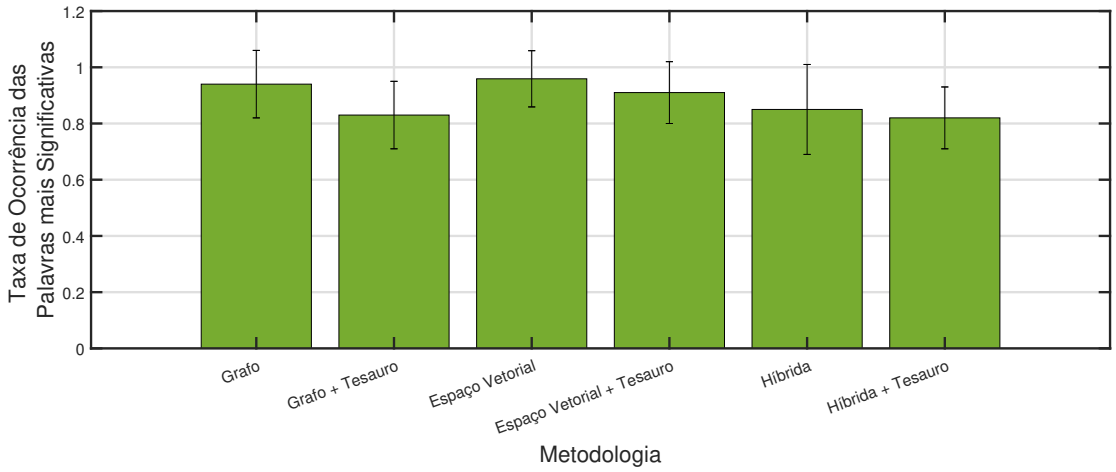


Figura 4.3: Avaliação do resumos gerados por seis abordagens diferentes. A Rezzumin aplica a abordagem híbrida com dicionário de sinônimos. A métrica mostra a proporção de palavras com os valores mais altos de TF-IDF que estão presentes no resumo gerado. A abordagem híbrida enriquecida com o tesauro apresenta desempenho um pouco menos adequado às palavras mais relevantes, pois considera mais tópicos e sinônimos no resumo gerado.

A segunda métrica de avaliação de resumos gerados, chamada **métrica da Proporção**, é uma contribuição deste trabalho. A lógica de construção compreende primeiramente calcular o vetor composto pela soma de todos os valores TF-IDF de cada palavra

de um texto. Em seguida, multiplica-se cada valor desse vetor pelo respectivo número de ocorrências de cada palavra pertencente ao resumo gerado. Aplicando a mesma lógica para o resumo de referência, adquiri-se uma versão ponderada do vetor de valores TF-IDF inicial, em que palavras mais significantes têm um peso maior. Finalmente, ao somar todas as ponderações vetoriais de cada resumo, encontra-se a proporção de palavras mais significantes entre os resumos dividindo um somatório pelo outro. A Figura 4.3 representa a razão de proporção do resumo gerado pelo resumo de referência. Observa-se que a relação em todas as metodologias é superior a 0,8, evidenciando que os resumos gerados detinham quase todas as palavras mais importantes dos textos analisados. O valor um pouco reduzido da abordagem híbrida se deve ao fato de esta elencar sentenças que cobrem mais tópicos no texto, mas que apresentam valores mais reduzidos na métrica TF-IDF, pois são menos frequentes no texto original. Além disso, o uso do tesauro também reduz essa métrica, pois implica na seleção de sentenças que também empregam sinônimos das palavras mais relevantes calculadas.

Capítulo 5

Criação de Estruturas Ontológicas usando Processamento de Linguagem Natural

Ontologia é uma das principais bases da Web Semântica. Uma definição formal e amplamente aceita que ontologia é uma especificação explícita de conceitualizações a respeito um domínio de interesse [65]. Assim, Aprendizado de Ontologia (OL) refere-se ao processo de construção de uma nova ontologia, ou aprimoramento de uma existente, a partir de uma fonte textual usando o mínimo de assistência de especialistas. Todo o processo de aprendizado de ontologia é constantemente associado a um "bolo de camadas". Tal analogia remete a pilha de tarefas que, à medida que são realizadas, geram saídas parciais representando uma camada do bolo. Essas saídas são diversas e podem incluir termos, conceitos, relações taxonômicas, relações não taxonômicas e, opcionalmente, axiomas [66].

Embora o processo possa ser podado de acordo com a necessidade do aplicativo, uma visão geral pode ser obtida primeiro extraindo termos de textos pré-processados usando uma associação de técnicas de processamento de linguagem natural (PLN) e medidas estatísticas ou probabilísticas. A tarefa de formar conceitos, a saída seguinte, envolve descobrir as variantes de um termo e agrupá-las por similaridade, através da implementação de análises de estrutura sintática e métodos de agrupamento. Ao modelar a interação entre os conceitos, é possível estabelecer dois tipos de relações, as taxonômicas, que seguem uma lógica hierárquica marcada pela presença de hiperônimos [1] ou hipônimos [2], e a relação não taxonômica que expressa um vínculo menos explícito, como a meronímia[3] ou relações de posse e causalidade. A última tarefa depende da descoberta de axiomas,

[1]Palavras cujos significados são mais abrangentes do que de outras.

[2]Palavras cujos significados são hierarquicamente mais específicos do que de outras.

[3]Relação semântica entre duas palavras, onde uma, o merônimo, significa algo que é parte do significado de outra, que significa um todo, o holónimo.

o que significa inferir uma regra de generalização ou dedução de um grande número de relações conhecidas que satisfazem critérios específicos.

Essa pluralidade no espectro de possíveis resultados de um procedimento de aprendizado ontológico, separa a ontologia em dois tipos: **ontologias leves**, marcadas por pouco ou nenhum uso de axiomas; e **ontologias pesadas**, marcadas pelo uso intensivo de axiomas. De uma perspectiva gráfica, as ontologias podem ser pensadas como grafos direcionados, em que conceitos são vistos como nós e relações como as arestas entre os nós.

Formalmente, existem quatro grandes frentes de ação para construir uma ontologia: (i) *abordagens baseadas na linguística* que prioriza o uso de técnicas de identificação de padrões sintáticos-lexicais, marcação de classes gramaticais (*part-of-speech*, POS) e análise de sentenças; (ii) *abordagens baseadas em estatísticas* que se concentram na análise de co-ocorrência e técnicas para poda ontológica e extração de associações e regras hierárquicas; (iii) *abordagens baseadas em lógica* que exploram teorias da programação em lógica indutiva e equivalências lógicas na tentativa de inferir ou derivar regras; (iv) *abordagens híbridas* que geralmente mesclam técnicas nativas de outros tipos de abordagens, sendo a abordagem predominante nos estudos existentes [4, 5, 67, 68].

5.1 A Abordagem Proposta para Geração de Taxonomias

Neste seção é apresentada uma abordagem de relacionamento sintático baseada em processamento de linguagem natural [69], aplicando algoritmos de agrupamento de forma eficiente para gerar taxonomias de conhecimento sobre documentos de domínio específico. A abordagem considera um estudo de caso de computação em nuvem por meio da coleta e análise de um conjunto de publicações recentes. Para validar a proposta, realiza-se uma avaliação dupla: (i) uma intra-metodológica comparando diferentes métricas intrínsecas de agrupamento e (ii) uma inter-metodológica na qual compara-se os resultados gerados de cobertura e popularidade com propostas do estado da arte.

A proposta descrita enquadra-se em uma abordagem híbrida, pois integra tanto elementos nativos da abordagem linguística, como a identificação de sintagmas nominais, quanto a abordagem estatística, por meio do uso de algoritmos de agrupamento. Conforme visto na Figura 5.1, a arquitetura da abordagem proposta é modular e permite a adição de novos módulos, bem como a adaptação a documentos de outras áreas. O

módulo **Coleta de Dados** adquire e converte arquivos de texto binários, expressos em figura como *Portable Document Format* (PDF), em texto simples. O módulo **Segmentação Textual** antecede as técnicas de processamento textual, garantindo a segregação de textos inteiros em um conjunto de frases. O módulo **Processamento de Linguagem Natural** (PLN) é implementado em linguagem `Python`, usando principalmente a biblioteca `NLTK` [4]. Tal módulo é responsável por pré-processar os dados, realizando tarefas de limpeza e ajuste dos dados textuais para então alimentar o módulo **Vetorização**. A vetorização transforma as palavras do texto em um espaço vetorial após construir um gráfico de conhecimento usando `NetworkX` [5]. Com os dados vetorizados, o módulo **Redução Dimensional** transforma os vetores de palavras em características úteis, que são tratadas pelo módulo **Algoritmos de Agrupamento**. Esses módulos usam funções fornecidas pelo `Scikit-learn` [6]. Por fim, os resultados são reunidos no módulo **Avaliação**, que realiza uma avaliação intrínseca e extrínseca, comparando com metodologias semelhantes.

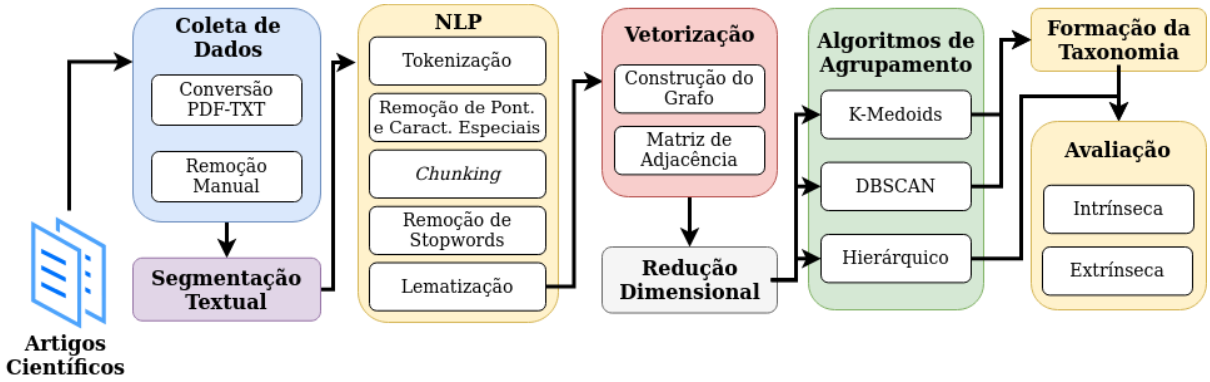


Figura 5.1: A arquitetura proposta integra um conjunto de módulos que proporcionam a construção e avaliação de uma taxonomia criada a partir de documentos de um domínio específico. Todo o processo começa com a coleta e processamento de textos extraídos de PDFs, identificando relações sintáticas específicas entre os termos. Em seguida, essas relações são vetorizadas antes de sofrer uma redução dimensional que privilegia as características mais importantes. Posteriormente, um conjunto triplo de algoritmos não supervisionados agrupa características semelhantes. Com base nos grupos criados, a taxonomia é finalmente construída e avaliada de acordo com métricas de comparação internas e externas.

Embora proponham formas inovadoras de criar taxonomias de domínio, nenhum trabalho prévio na literatura apresenta uma avaliação quantitativa aprofundada ou mesmo utiliza em seu desenvolvimento textos totalmente extraídos de artigos. Outro avanço perante às abordagens do estado da arte, está no fato de a proposta apresentada extrair

[4]Disponível em: https://www.nltk.org/.
[5]Disponível em: http://networkx.lanl.gov/.
[6]Disponível em: https://scikit-learn.org/stable/.

as palavras mais significativas diretamente de dados textuais não estruturados, criando posteriormente uma representação em grafos da correlação entre os termos. O gráfo do conhecimento extraído dos documentos textuais considera a relação sintática entre os termos, extraída por meio de técnicas de processamento de linguagem natural, bem como a importância de cada termo calculada por algoritmos de aprendizado de máquina não supervisionados.

5.2 Coleta de Documentos e Pré-Processamento

A estratégia de coleta de documentos adotada inclui a aquisição de 82 artigos científicos publicados na Conferência IEEE Globecom 2019 [7], em todos contendo palavras-chave com alguma variação do termo "computação em nuvem". Antes dos procedimentos linguísticos de PLN, é necessário extrair o conteúdo dos arquivos PDF. Esta tarefa é realizada por uma abordagem semi-automática, primeiro usando o `PDF-Layout-Scanner` [8], uma biblioteca em `Python`, cuja função é converter um arquivo PDF de duas colunas em arquivos de texto. Em uma segunda etapa manual, todos os elementos não textuais são removidos, excluindo assim tabelas, referências bibliográficas e fórmulas. Vale ressaltar que, como a grande maioria dos trabalhos da área, o ponto final foi o caráter delimitador adotado no processo de subdivisão do texto contíguo em frases.

O tratamento textual foi realizado aplicando a *tokenização*, remoção de pontuação e caracteres especiais, eliminação de *stopwords* e *lematização*, correspondendo respectivamente às etapas 1, 2, 3, e 6b descritas na Seção 2.1. Entre as etapas de remoção de pontuação e eliminação de *stopwords*, realiza-se uma etapa intermediária denominada *chunking* com o objetivo de selecionar relações sintáticas específicas no texto. Basicamente, a *chunking*, também chamada de análise sintática superficial, analisa toda a frase, primeiro identificando as partes constituintes das frases (substantivos, verbos, adjetivos) e depois ligando-as a unidades de ordem superior com significado gramatical discreto. Em particular, o objetivo desta tarefa é selecionar em cada frase apenas as estruturas sintáticas caracterizadas pela união de dois sintagmas nominais por um verbo, seja em qualquer conjugação. Sintagmas nominais são frases cujo núcleo é um substantivo.

[7] Disponível em: https://ieeexplore.ieee.org/xpl/conhome/8968653/proceeding.
[8] Disponível em: https://pypi.org/project/PDF-Layout-Scanner/

5.3 Vetorização e Redução Dimensional

Após o tratamento textual na Seção 5.2, compreende-se cada frase remanescente como uma lista de lemas contendo os elementos mais significativos: par de sintagmas nominais ligados por um verbo. A partir dessas associações gramaticais, é possível construir um grafo do conhecimento, no qual pares de nós representam os sintagmas nominais, enquanto a aresta entre eles representa o verbo. Para obter uma representação matematicamente operável do grafo de conhecimento criado, usa-se a matriz de adjacência. Nessa matriz quadrada binária, cada elemento indica se dois vértices são adjacentes ou não no grafo. Como a dimensão da matriz é determinada diretamente pelo número de nós presentes no grafo, no caso 6047 nós, aplica-se a indexação semântica latente. Com isso, o número de características foi reduzido para 5000, correspondendo a uma redução de 26,4% do espaço de memória, enquanto preservou cerca de 99,1% da variância das características originais. A fração de variância subtraída é aproximadamente uma medida da quantidade de informações descartada após o processo de redução de dimensionalidade.

5.4 Aplicação de Algoritmos de Agrupamento

A partir de uma matriz de características mais compacta, foi possível agrupá-las usando cada tipo de algoritmo não supervisionado descrito na Seção 2.5.2, sendo eles: *k-medoids*, DBSCAN, hierárquico aglomerativo. Uma vez que a escolha do parâmetro k, número de agrupamentos, precede o uso do algoritmo *k-medoids*, aplicam-se os métodos *Elbow* e da Silhueta sobre a matriz de características. Na Figura 5.2, observa-se que os métodos analisam a coesão e separação dos agrupamentos numa faixa de 100 a 5000 agrupamentos. Contudo, neste intervalo não é possível determinar com precisão o ponto em que a curva *Elbow* decresce abruptamente. Nessas circunstâncias, a maneira de aplicar *k-medoids* aos dados textuais usados, é definir k igual a 3000, pois é um mínimo local na curva do método *Elbow* e um máximo local na curva da Silhueta. Esta escolha permite compactar os 5000 características fornecidos pela LSI em um conjunto de 3000 dimensões artificiais. Tal valor de k foi igualmente adotado pelo algoritmo hierárquico.

Paralelamente, como o algoritmo DBSCAN não permite a escolha explícita do número de agrupamentos, foi realizado um procedimento de varredura, com o objetivo de testar diferentes combinações dos parâmetros ϵ e $minPts$. Para obter um resultado que divide a matriz reduzida de características em 3.000 agrupamentos, adota-se $\epsilon = 1,029$ e $minPts = 1$.

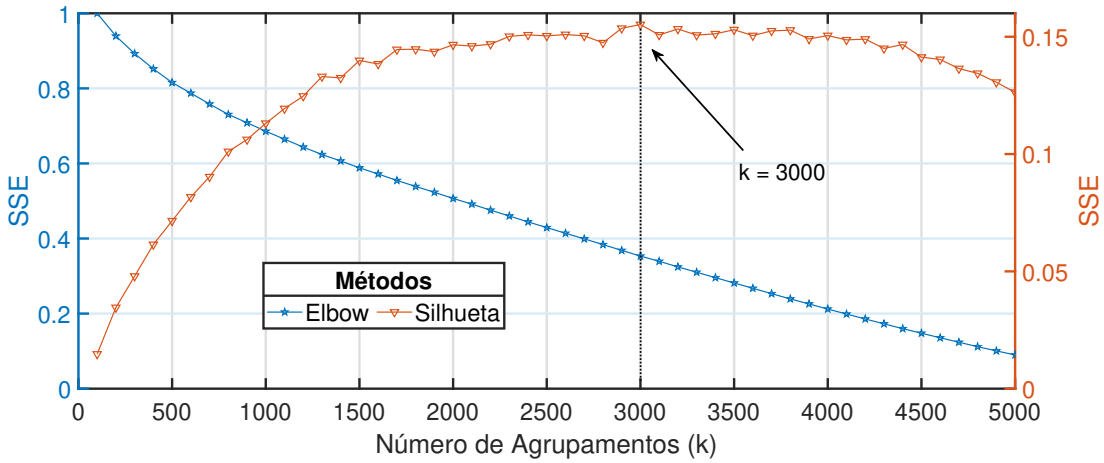


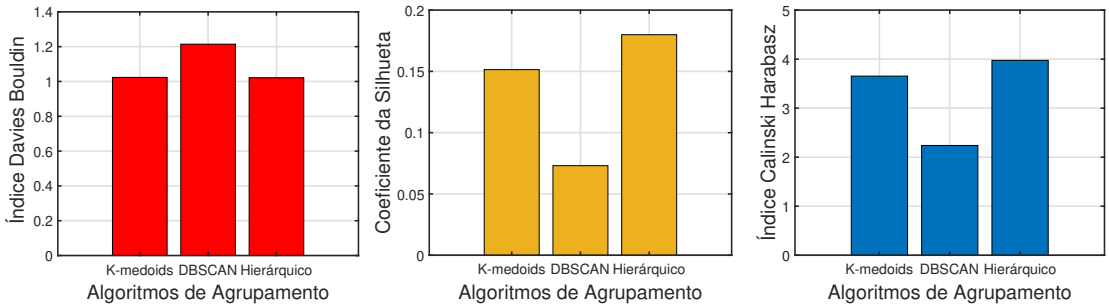
Figura 5.2: Métodos para encontrar o melhor número de agrupamentos k (*clusters*). Para ambos os métodos, os valores de erro quadrático médio normalizado são mostrados. Para a faixa de valores testados para k, 3.000 agrupamentos mostra-se o mais adequados, mesmo não tendo o ponto de quina evidente na curva do método *Elbow*.

5.5 Avaliação da Proposta

A falta de um banco de dados de referência, implica a adoção de formas alternativas de avaliação, além das conhecidas métricas de recuperação de informações ou mesmo com base em medidas extrínsecas. Portanto, o objetivo da avaliação limita-se a duas abordagens possíveis, uma intra e outra inter-metodológica.

5.5.1 Avaliação Intra-Metodológica

O objetivo da avaliação é comparar as metodologias de agrupamento, desenvolvidas na presente proposta, através de três medidas intrínsecas descritas na Seção 2.5.3, índice Davies Bouldin, Coeficiente da Silhueta e índice Calinski Harabasz. A Figura 5.3 mostra a comparação interna entre as qualidades dos três algoritmos de agrupamento empregados. Analisando os valores apresentados por cada algoritmo, é notório por apontar um consenso de que os resultados do agrupamento hierárquico foram os melhores. Entre os algoritmos testados, este foi unânime em atingir o menor índice Davies Bouldin, traduzido em uma densidade mais baixa, e as maiores pontuações do Coeficiente de Silhueta e Calinski Harabasz.



(a) Davies Bouldin. (b) Coeficiente da Silhueta. (c) Calinski Harabasz.

Figura 5.3: Os resultados obtidos na avaliação interna consideram três métricas que quantificam a qualidade dos agrupamentos criados por três algoritmos não supervisionados. Os melhores resultados são encontrados usando um algoritmo de agrupamento hierárquico.

5.5.2 Avaliação Inter-Metodológica

Em uma avaliação inter-metodológica, os resultados da metologia proposta são comparados com aqueles encontrados usando a metodologia de Casagrande *et al.* [1]. Sua metodologia, originalmente testada no campo de pesquisa de *smart grid*, fornece uma construção de taxonomia usando três etapas principais: coleta de dados, classificação de palavras-chave e geração de taxonomia. A primeira etapa consiste na formação de um banco de dados composto por diversas publicações relevantes em um domínio específico do conhecimento. Assim, o resumo e o conjunto de palavras-chave são extraídos de cada documento. O segundo estágio compreende a construção de um grafo bidirecional em que cada palavra-chave representa um nó, e o peso de cada aresta é calculado de acordo com uma medida de similaridade propositalmente assimétrica. Tal métrica é definida na equação

$$NGD_M(x, y) = \alpha * NGD_A(x, y) + (1 - \alpha) * NGD(x, y), \tag{5.1}$$

como uma ponderação entre a tradicional **Distância Normalizada do Google** (*Normalized Google Distance*, NGD) e uma versão modificada dela que considera o conjunto de resumos, a NGD_A. Como não existe uma definição explícita do coeficiente de ponderação α em [1], adotamos $\alpha = 0,3$. A tradicional NGD [70] é uma forma de medir a distância semântica entre dois termos por meio do número de resultados retornados pelo mecanismo de busca do *Google*. Matematicamente, NGD é expressa como

$$NGD(x, y) = \frac{max\{log\, f(x), log\, f(y)\} - log\, f(x, y)}{log\, N - min\{log\, f(x), log\, f(y)\}} \tag{5.2}$$

em que f_x e f_y são o número de resultados para os termos de pesquisa x e y, respecti-

vamente, e $f_{x,y}$ é o número de páginas da *web* em que x e y estão presentes. Embora a escolha do parâmetro N não seja fixa, é aconselhável usar um valor grande. Em particular, no presente trabalho adota-se $N = 25270000000$, o número de resultados encontrados ao pesquisar qualquer um dos artigos "the" ou "a" no *Google*.

Por outro lado, NGD_A realiza o mesmo procedimento, mas restringe o domínio de busca de palavras-chave, contabilizando somente os resultados obtidos por meio de uma busca interna nos textos dos resumos. A forma assimétrica desta medida de similaridade se destaca ao observar uma diferença real nos resultados obtidos pelo cálculo da similaridade entre a palavra x e a palavra y e a similaridade da palavra y e a palavra x.

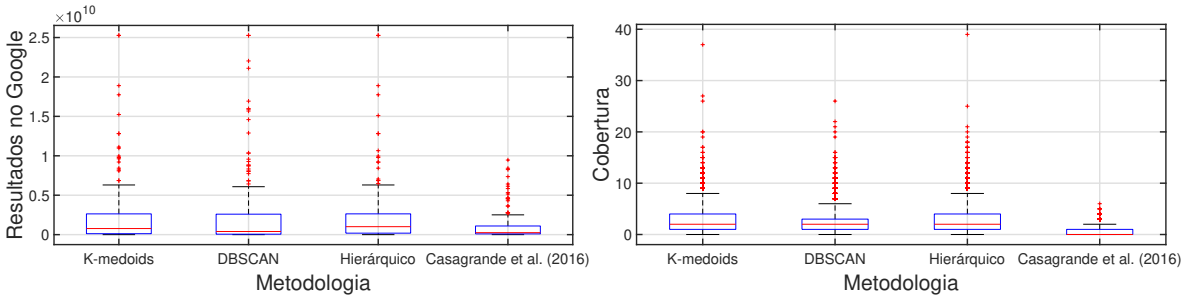
Uma vez construído o grafo, torna-se possível classificar cada nó, ou seja, cada palavra-chave, de acordo com sua centralidade de proximidade. A terceira e última etapa emprega a classificação de proximidade anterior, organizada de forma descendente, na aplicação do algoritmo de Heymann [71] que padroniza a construção da taxonomia. O algoritmo começa com uma árvore de nó único, a raiz sendo a primeira palavra-chave na classificação da proximidade. Em seguida, o algoritmo prossegue adicionando cada palavra-chave à árvore de cada vez, seguindo a ordem de classificação de centralidade. A decisão sobre onde colocar cada palavra-chave é guiada pelo cálculo de sua similaridade do cosseno com cada nó atualmente presente na árvore, encontrando assim o nó mais semelhante. A palavra-chave é então adicionada como um nó filho do nó mais semelhante se a sua semelhança com este nó for superior a algum limite σ ou é adicionada como um nó filho do nó raiz se for inferior ao σ estabelecido.

Reproduzindo os mesmos procedimentos mencionados anteriormente, mas introduzindo o *corpora* relacionado a computação em nuvem coletada, é possível estabelecer uma base adequada de comparação com os resultados da abordagem proposta. Vale ressaltar que para o σ adotado no processo de reprodução da abordagem concorrente, $\sigma = 0,1$, foi possível identificar ao final, a presença de 177 palavras-chave na taxonomia gerada. A escolha desse valor σ foi baseada em uma queda considerável observada no valor da similaridade entre palavras-chave não relacionadas.

Para fins de equidade da avaliação inter-metodológica, no âmbito quantitativo, ambas as metodologias utilizam o mesmo número de palavras-chave. Para compor a taxonomia, foi realizado uma seleção das palavras-chave mais importantes em dois ramos da metodologia proposta, os ramos utilizando os algoritmos *k-medoids* e o DBSCAN. Tal ação é desnecessária no ramo de agrupamento hierárquico visto que a própria execução do algoritmo já organiza as palavras-chave segundo uma disposição em níveis. Embora o

critério mais direto para escolher as 177 palavras-chave mais significativas seja escolher as mais centrais, isso possivelmente resultaria em ambiguidades, pois alguns algoritmos de agrupamento não garantem a convexidade dos agrupamentos gerados. Portanto, o critério adotado foi obter todas as palavras-chave pertencentes aos agrupamentos mais densos até atingir o número desejado, 177.

A Figura 5.4(a) mostra os resultados da primeira métrica de comparação que buscou avaliar a variação no número de acessos no *Google* para cada palavra-chave associada às metodologias. Tal resultado pode ser interpretado como um grau de popularidade do conjunto de palavras pesquisadas, ou até mesmo seu estado atual de maturidade no campo da computação em nuvem. Notar-se que todas as abordagens propostas obtiveram resultados de popularidade melhores do que aqueles obtidos pela metodologia de Casagrande *et al.*, tanto em uma perspectiva de comparação por média quanto por pico.



(a) Comparação da Po. (b) Comparação da Cobertura

Figura 5.4: Os resultados obtidos na avaliação externa comparando a metodologia tripla proposta e a metodologia de Casagrande *et al.* [1]. Observa-se a superioridade em todos os cenários desenvolvidos da metodologia proposta, onde destaca-se a popularidade e cobertura de até 2,5 e 6 vezes maior respectivamente, quando comparada com uma metodologia do estado-da-arte.

Também avalia-se o nível de cobertura de cada metodologia. Essa medida expressa a razão entre a soma das ocorrências do conjunto de palavras-chave em cada frase do *corpora* e o número total de palavras-chave na taxonomia gerada. A Figura 5.4(b) destaca que os resultados das abordagens proposta novamente foram superiores aos da proposta de Casagrande *et al.*. Em particular, destaca-se a alta taxa de cobertura alcançada pelo ramo hierárquico, refletindo uma taxonomia mais informativa. Um aspecto curioso revelado na avaliação foi a presença significativa de termos relacionados à inteligência artificial nas taxonomias geradas, levando a crer que este campo estará possivelmente associado à próxima geração da computação em nuvem.

Capítulo 6

Detecção de Notícias Falsas

Em 2016, durante as eleições presidenciais dos Estados Unidos, a sociedade americana testemunhou uma epidemia alarmante de notícias falsas, cujos efeitos foram sentidos multilateralmente. Efeito semelhante foi sentido nas eleições de 2018 no Brasil. Devido ao seu potencial de disseminação, aceitação e destruição [72], as notícias falsas são atualmente uma das grandes ameaças ao conceito de verdade lógica, deteriorando a democracia, o jornalismo, a justiça e até a economia [12, 55]. Esta última, em especial teve de lidar com flutuações de 130 bilhões na bolsa de valores, como consequência de uma declaração falsa afirmando que Barack Obama havia se ferido em uma explosão[1]. Recentemente em meio uma crise sanitária global, um estudo revelou que o Brasil está entre os 7 países com maior fluxo de disseminação de informações falsas a respeito da Covid-19 [73]. Há um crescente esforço conjunto da comunidade acadêmica para desenvolver abordagens capazes de analisar, detectar e intervir na atuação desses conteúdos enganosos. Comprovações científicas já revelaram a vulnerabilidade dos humanos em distinguir verdade e falsidade, sendo reduzida a quase uma probabilidade aleatória, em média 54% de acerto [12, 55, 74, 52].

Apesar da inexistência de um consenso claro sobre o conceito de notícias falsas (*fake news*), uma das definições formais mais aceitas as interpreta como notícias intencionalmente e verificavelmente falsas. Com relação a tal definição, destacam-se dois aspectos: a intenção e a autenticidade. O primeiro aspecto diz respeito à intenção desonesta usada com o intuito de enganar o leitor. Já o segundo se relaciona com a possibilidade de essas informações falsas terem sua veracidade checada. Uma vantagem dessa definição é a eliminação da ambiguidade gerada por diversos conceitos que frequentemente co-ocorrem e

[1]Disponível em: https://www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/#559102f12fac.

se sobrepõem ao conceito de notícias falsas. Uma síntese desses múltiplos conceitos, não considerados notícias falsas segundo essa definição, pode ser elencada como: (i) sátiras e paródias, que pelo conteúdo humorístico embutido, usando sarcasmos e ironias, é factível de ter seu caráter enganoso identificado; (ii) rumores e boatos, que não se originaram de eventos de notícias, porém são aceitos publicamente; (iii) teorias de conspiração, por não serem facilmente verificáveis como verdadeiras ou falsas; (iv) desinformação que é criada involuntariamente, sem uma origem ou intenção específica; e (v) trotes e embustes (*hoaxes*) que são motivados apenas por diversão ou para enganar indivíduos direcionados [75, 76, 12].

Em uma perspectiva psicológica e social, é possível diagnosticar fatores que contribuem para a produção e disseminação das notícias falsas. Na psicologia, esses fatores são identificados como vulnerabilidades individuais conhecidos como **realismo ingênuo**, em que consumidores tendem a acreditar que suas percepções da realidade são os únicos pontos de vista, enquanto as demais são consideradas desinformadas, irracionais ou tendenciosas; e **confirmação de tendência**, que explica a preferência dos consumidores de receber informações que confirmem suas visões existentes. Considerando o campo social, a disseminação das notícias falsas está intimamente ligada com a dinâmica social dos indivíduos, podendo ser fundamentada em três teorias: a Teoria da Prospecção, que descreve a tomada de decisão como um processo pelo qual os indivíduos fazem escolhas com base nos ganhos e perdas relativas em comparação com seu estado atual; a Teoria da Identidade Social e a Teoria da Influência Normativa, na qual enfatizam que aceitação e afirmação social são essenciais para a identidade e autoestima de um indivíduo, fazendo com que os usuários escolham ser "socialmente seguros" [76].

Embora a existência de notícias falsas preceda o surgimento das das mídias sociais, seu advento alterou e ampliou a dinâmica de propagação desse tipo de informação, inclusive adicionando novos atuantes como *social bots*, *trolls*, *ciborgs*. Todas essas contas maliciosas altamente ativas e partidárias têm um único propósito: tornarem-se fontes poderosas de proliferação de notícias falsas. Outro fator atual que facilita a disseminação desse tipo de notícia é o fenômeno de bolha social (*echo chamber*) em que usuários tendem a se relacionar virtualmente com seus *like-minders*, ou seja, pessoas que pensam como eles. Nessas bolhas sociais estão presentes duas ideias principais, sendo a primeira conhecida como credibilidade social. Tal ideia é explicada pelo fato de as pessoas serem mais propensas a considerar uma fonte como credível se os outros também a considerarem, especialmente quando não há como se comprovar. A segunda ideia remete a uma heurística de frequência, segundo a qual consumidores naturalmente preferem notícias que são ouvidas mais

constantemente, mesmo sendo falsas [76].

Estudos prévios propõem diferentes teorias analíticas que são potencialmente úteis na contenção de notícias falsas. Primeiramente, tem-se a análise baseada no estilo da escrita, a qual trabalha com premissa de que as notícias falsas detêm perfis de escrita únicos, totalmente diferentes dos seus pares legítimos [77]. Paralelamente, tem-se a análise baseada na propagação, mapeando o grau de espalhamento a partir de modelos epidêmicos adaptados da biologia [78]. Por último, tem-se a análise baseada no usuário, que considera o seu papel na disseminação das notícias, consequentemente distinguindo um usuário malicioso daqueles sem má intenção [79].

6.1 A Abordagem Estilístico-Computacional Proposta

Esta seção introduz uma abordagem baseada numa análise estilístico-computacional, fundamentada no processamento de linguagem natural, que aplica eficientemente algoritmos de aprendizagem de máquina na detecção de notícias falsas em textos extraídos de mídias sociais [80]. A hipótese inicial remete à ideia de que tanto as notícias falsas quanto as legítimas detêm perfis dispares no formato da escrita. O enfoque na abordagem da análise estilístico-computacional justifica-se pelo fato de que o usuário que consome mídias sociais possui acesso somente ao conteúdo da informação e não aos modelos de propagação ou aos perfis de usuários compartilhadores. Abordagens usando o modelo de propagação e o modelo de perfis de usuário compartilhadores são vantajosas quando aplicadas por provedores de serviço e provedores de conteúdo.

A arquitetura da abordagem estilístico-computacional proposta emprega três metodologias distintas para classificar a legitimidade de uma notícia. Como expresso na Figura 6.1, as duas primeiras metodologias implementam diferentes combinações de algoritmos de aprendizado de máquina, tanto de agrupamento não-supervisionado (clusterização) quanto classificação, na tentativa de prever o tipo da notícia a partir do treinamento somente sobre as notícias verdadeiras. Paralelamente, a terceira metodologia expande a proposta de detecção para um cenário estatístico, partindo da hipótese de que notícias verdadeiras e falsas têm distribuição de probabilidades distintas ao se considerar o módulo de sua representação no espaço vetorial de frequência de palavras.

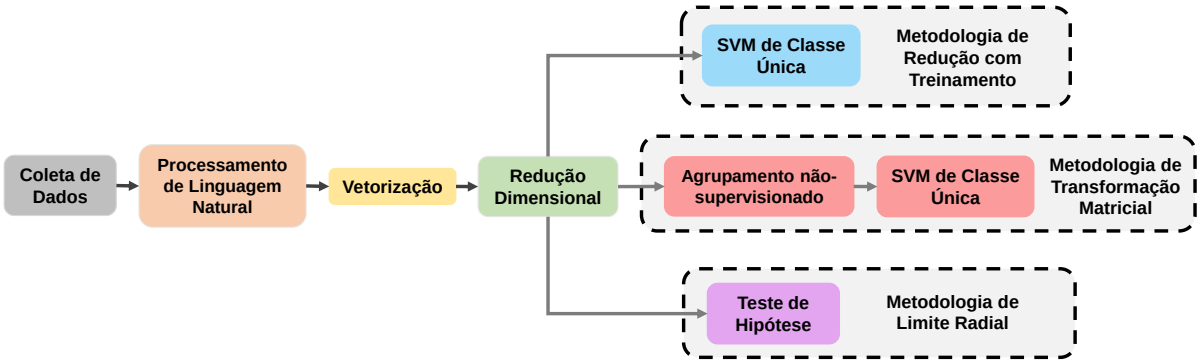


Figura 6.1: A arquitetura de detecção proposta inicia com a coleta de notícias seguida do processamento textual e vetorização dos textos. Após uma redução dimensional, são testadas três metodologias para detecção de notícias falsas.

6.2 *Web Scraping* e Construção da Base de Dados

A construção de uma base de dados com qualidade e disponibilidade é o pilar de qualquer mecanismo automático de detecção de notícias falsas. Sua importância está atrelada à necessidade de armazenar a máxima quantidade de exemplos contrastantes, notícias falsas e verdadeiras, para então serem absorvidos e treinados por algoritmos de aprendizado [81]. Considerando a análise do conteúdo das diferentes bases de dados existentes atualmente dedicadas a detecção de notícias falsas, classificam-se em três principais grupos:

- **declarações curtas** provenientes tanto de sites dedicados a checagem de fatos, como PolitiFact[2], Channel4.com[3], Snopes[4], que concentram declarações de debates, campanhas e redes sociais manualmente coletadas e rotuladas por editores e jornalistas [55], quanto de sites gerais, como a Wikipedia, em que houve uso de anotadores humanos na classificação [82].

- **postagens de redes sociais**, principalmente do *Facebook* e do *Twitter*. Um exemplo desse tipo ocorre em BuzzFace [83], em que a BuzzFeedNews[5], uma base de dados extraída de postagens no *Facebook*, é extendida adicionando comentários relacionados as postagens presentes nela. Exemplos relativos ao *Twitter* são vistos em PHEME [84] e CREDBANK [85] que extraem e classificam os *tweets* segundo uma rotulagem binária (verdadeiro ou falso) baseando-se nas relações entre seguidores; ou usando uma concatenação de pontuações sobre a veracidade do *tweet* fornecidas por 30 anotadores humanos.

[2]Disponível em: https://www.politifact.com/
[3]Disponível em: https://www.channel4.com/news/factcheck
[4]Disponível em: https://www.snopes.com/fact-check/
[5]Disponível em: https://github.com/BuzzFeedNews/2016-10-facebook-fact-check

- **artigos inteiros** cujos exemplos mais notórios são FAKENEWSNET [86] e BS DETECTOR[6], FullFacts[7]. Limitando-se a língua portuguesa, o Fake.Br Corpus [56] é composto por uma quantidade simétrica de notícias comprovadamente falsas e por notícias verdadeiras, contendo não somente os textos analisados mas também meta-dados relacionados.

Nesse contexto, uma eventual coleta errônea de dados tem o potencial de causar inúmeras consequências negativas, que variam desde a particularização da análise até a obtenção de resultados dissonantes. Logo, é prudente adotar algumas diretrizes sugeridas por Rubin *et al.* para a formação de um *corpus* de notícias falsas [75]. Rubin *et al.* defendem que qualquer construção de uma base de dados, *corpus*, de notícias falsas deve se ater a nove condições importantes, a saber. (i) Considerar tanto as instâncias falsas como as verdadeiras permite que eventuais métodos preditivos aplicados à base considerem padrões característicos de cada tipo de notícia. (ii) A informação deve estar preferencialmente em formato textual, em vez de ser apresentada como mídia em formato de áudio ou vídeo. Informações nesses formatos devem ser transcritas, tornando-se manipuláveis por ferramentas de processamento de linguagem natural. (iii) A homogeneidade das notícias quanto ao tamanho e (iv) quanto a maneira da escrita, são outras duas condições a serem consideradas, evitando sempre que possível instâncias muito díspares. Igualmente, existe uma preocupação com (v) a forma de distribuição das notícias, visto que há suspeitas de que ao saber como e em qual contexto estas foram fornecidas, *e.g.* humorístico, sensacionalista, pode-se influenciar os leitores. Além disso, (vi) a aquisição de notícias de um mesmo intervalo temporal é um fator primordial, pois os assuntos podem variar drasticamente em um curto intervalo de tempo. Adicionalmente, (vii) é aconselhável atender a alguns aspectos pragmáticos, tais como custos com direito autoral, disponibilidade, facilidade de obtenção e privacidade dos escritores. Não se deve negligenciar o (viii) idioma e a (ix) cultura a que pertencem os dados coletados, pois a tradução pode implicar ambiguidades ou más interpretações, afetando negativamente a eficiência de processos de detecção [75, 87].

Sendo assim, a composição da base de dados inclui tanto notícias verdadeiras quanto falsas, coletadas a partir de contas específicas do *Twitter*. Tal procedimento de extração de dados armazenados em sítios *web* é denominado *web scraping*. A fim de obter essas informações, foi desenvolvido um *script* em `Python` empregando a interface de programação de aplicação (*Application Programming Interface* - API) do *Twitter*. O acesso a essa API,

[6]Disponível em: https://github.com/bs-detector/bs-detector
[7]Disponível em: https://fullfact.org/

usando credenciais de desenvolvedor, permite a extração contínua do conteúdo textual dos *tweets* de qualquer perfil aberto na rede social. Contudo, além das limitações temporais igualmente enfrentadas por Barreto *et al.*, como o número máximo de requisições por janela de tempo de 15 minutos [57], há também limitação da quantidade de *tweets* históricos passíveis de serem coletados. Dessa maneira, a obtenção de *tweets* é restrita a um período de até, no máximo, dois meses passados a contar da data de execução do *script*.

Devido às restrições de recuperação de dados, uma solução é diversificar as fontes de busca por notícias verdadeiras, coletando *tweets* de outras fontes jornalísticas. A escolha de perfis de veículos jornalísticos como fonte de conteúdo verdadeiro parte da premissa que estes perfis são menos susceptíveis a compartilhar conteúdo de procedência duvidosa do que contas de usuários individuais. Analogamente, promove-se também a coleta de *tweets* comprovadamente falsos, previamente verificados por jornalistas e disponibilizados pelo perfil "Boatos.org". Assim, a base de dados gerada é composta por 33 mil *tweets*, sendo detalhada na Tabela 6.1.

Tabela 6.1: Composição da base de dados de notícias

Tipo	**Veículo de Mídia**	**Número de Tweets**	**Intervalo Temporal**
Verdadeiro	BBC Brasil	3158	01/10/19 - 30/10/19
	Jornal Estadão	3120	04/10/19 - 30/10/19
	Jornal O Globo	3208	24/09/19 - 29/10/19
	Jornal Extra	3244	14/09/19 - 30/10/19
	Jornal Folha de São Paulo	2787	11/10/19 - 29/10/19
	Revista Veja	3187	12/09/19 - 31/10/19
	Portal G1	3181	10/09/19 - 27/10/19
	Portal R7	2897	26/09/19 - 30/10/19
	Revista Exame	2787	02/10/19 - 30/10/19
	UOL	3130	09/09/19 - 31/10/19
Falso	Boatos.org	3180	21/09/19 - 30/10/19
Total		**33873**	

6.3 Limpeza e Conformação dos Dados com Processamento de Linguagem Natural

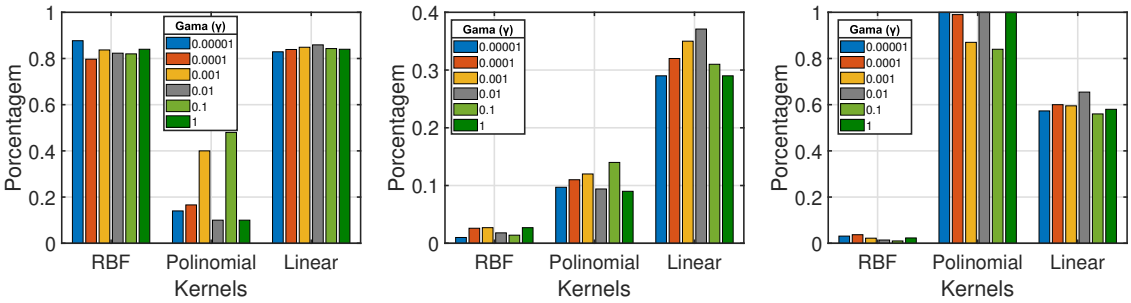
Depois de consolidar a base de dados, cada *tweet* é limpo e conformado empregando as etapas 1-6a do PLN, sendo elas respectivamente *tokenização*, remoção de pontuação e caracteres especiais, eliminação de *stopwords*, correção ortográfica, reconhecimento de

entidades nomeadas e *stemização*. Tratando-se especificamente de sentenças extraídas do *Twitter*, torna-se igualmente importante a remoção de vocábulos próprios da plataforma, como eventuais *hashtags*, indicações ou *links* de compartilhamento. Uma vez processada textualmente, a base de dados inteira é traduzida na sua representação vetorial usando o modelo TF-IDF. Imediatamente após esse procedimento de vetorização, identifica-se 13,716 radicais distintos o que ascendeu a necessidade por uma redução dimensional a fim de tornar os procedimentos seguintes menos custosos. Assim com a preservação de aproximadamente 70% da variância das características originais, a Indexação Semântica Latente reduziu o número de características para 2.000, correspondendo uma economia de 85% no espaço de memória.

6.4 A Metodologia de Redução com Treinamento

O uso de algoritmos supervisionados para a detecção de notícias falsas depende de uma grande base de dados contendo tanto notícias verdadeiras, como falsas. Contudo, isso impõe a limitação de haver uma base rotulada com notícias reais e falsas. As notícias falsas, embora sejam cada vez mais numerosas, são difusas nas redes sociais e tendem a ser voláteis, já que algum período após a disseminação perdem a credibilidade. Uma estratégia para contrapor a limitação no número de notícias falsas para o treinamento dos classificadores é o aprendizado de uma única classe, como o baseado no algoritmo máquina de vetor de suporte de classe única (*One-class Suport Vector Machine*). A SVM de classe única é um algoritmo de aprendizado não supervisionado que deriva um hiperplano de decisão para detecção de anomalias. Novos dados são classificados como semelhantes ou diferentes do conjunto de treinamento. Em contraste com as implementações típicas da SVM, a classe única leva em consideração um conjunto de amostras de treinamento de uma única classe. Qualquer nova amostra que não se encaixe na superfície de decisão definida pelo conjunto de treinamento é considerada uma instância de uma nova classe e, portanto, uma notícia falsa [88, 89]. O processo de aprendizagem usando SVM de classe única conta com o emprego da função de núcleo (*kernel*) linear.

Os resultados apresentados contam com a base de dados particionada em 90% para treinamento e 10% para o teste. Vale ressaltar também que uso da SVM de classe única como modelo de classificador impõe um requisito singular: a etapa de treinamento deve ser realizada usando apenas as instâncias de notícias verdadeiras, a classe única. A Figura 6.2 mostra a execução do classificador de classe única para diferentes funções de núcleo e variando o coeficiente γ que determina o quanto amostras distantes influenciam no cálculo



(a) Acurácia. (b) Precisão. (c) Sensibilidade.

Figura 6.2: Resultados obtidos aplicando o LSI juntamente com a SVM de classe única. A melhor acurácia é encontrada usando função de núcleo linear e $\gamma = 0{,}01$.

do hiperplano da SVM. Vale ressaltar que os melhores resultados de acurácia e precisão para a classe de notícias falsas são obtidos usando a função de núcleo linear e $\gamma = 0{,}01$.

6.4.1 A Metodologia de Transformação Matricial

Após a redução dimensional, a matriz M, que comporta todas as amostras da base de dados, possui um número de colunas j igual a 2.000, equivalente à quantidade de características restantes em cada amostra. Ao supor a existência de uma matriz transformada T, de dimensão $j \times i$, tal que i seja bem menor que j, pode-se afirmar que sua projeção pela matriz D resulte numa matriz S ainda mais compacta. Na prática, a obtenção da matriz transformada consiste em:

$$D_{k\times i} \times T_{i\times j} = S_{k\times j}. \tag{6.1}$$

Supondo a existência de uma possível matriz de transformação, formada pelos centroides que melhor descrevem os dados na matriz de características M, aplica-se a esta matriz o algoritmo *k-means*. Analisando a Figura 6.3, a qual concentra os resultados para determinação de k obtidos pelos métodos *Elbow* e da Silhueta descritos na Seção 2.5.2.1, a melhor maneira de aplicar o *k-means* é definindo k igual a 56. Tal conclusão, apesar de não ser a ideal, é a que simultaneamente apresenta um mínimo local na curva *Elbow* e um máximo global na curva da Silhueta. A escolha de $k = 56$ permite definir 56 centroides, formando a matriz $T_{i\times j}$ que compacta as 2.000 características providas pela LSI em um conjunto de 56 dimensões artificiais. Sobre a matriz $S_{k\times j}$, aplica-se então o classificador SVM de classe única. A abordagem de usar a transformação para o espaço vetorial definido pelos k centroides tem o objetivo de concentrar os dados, pois mesmo após a aplicação do LSI, a matriz M é esparsa.

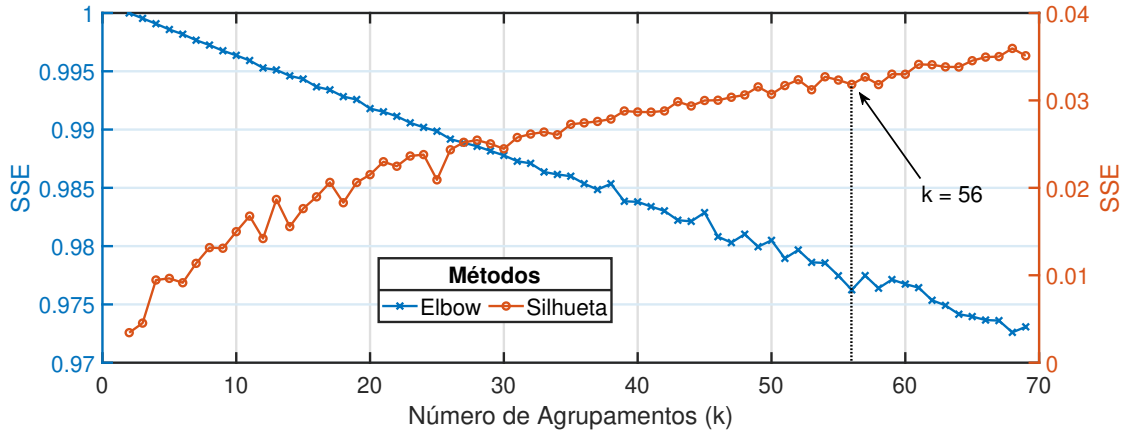


Figura 6.3: Métodos para determinação do melhor número de agrupamentos k (*clusters*). Para ambos os métodos são mostrados os valores do erro médio quadrático normalizados. Para os valores testados para k, 56 agrupamentos se mostrou como um mínimo local para a curva *Elbow* e um máximo local para curva da Silhueta.

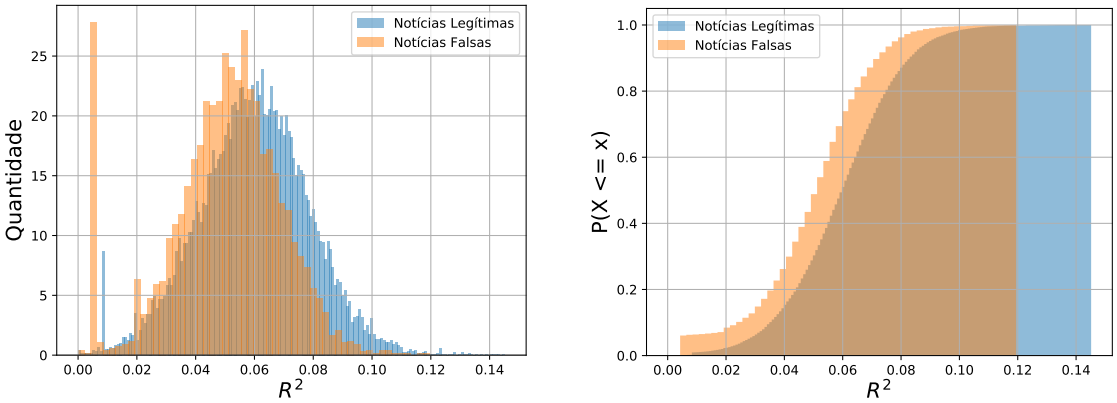
6.4.2 A Metodologia de Limite Radial

A abordagem de limite radial parte da matriz M de características, com 2.000 características, resultante da redução dimensional pelo LSI, e implementa a Equação 6.2.

$$R_i^2 = x_{i0}^2 + x_{i1}^2 + x_{i2}^2 + ... + x_{ik}^2 \tag{6.2}$$

Esta equação define que o somatório de todas os valores x_{ij}^2, pertencentes a uma amostra, linha i da matriz M, resulta em uma variável aleatória R_i^2, que pode ser interpretada como o raio de uma hiperesfera que contém as notícias verdadeiras. Expandindo essa lógica para toda a matriz, porém agrupando por tipo de notícia, falsa e legítima, obtém-se um conjunto de variáveis aleatórias próprias de cada tipo de notícia. Em posse desse conjunto, as Figuras 6.4(a) e 6.4(b) representam a função de densidade de probabilidade e de distribuição acumulada de notícias verdadeiras e falsas.

Observando a Figura 6.4(a), cogita-se a hipótese de desigualdade entre as médias referentes aos conjuntos de variáveis aleatórias de notícias falsas e de legítimas. A comprovação dessa suposição, refutando a hipótese nula com um um intervalo de confiança de 95%, surge através da aplicação do teste t de Welch, uma adaptação do teste t de Student para conjuntos de amostras com variância ou tamanhos distintos. Explorando essa relação estatística entre os conjuntos de notícias e valendo-se do fato da distribuição t de Student se aproximar da distribuição normal para graus de liberdade elevados, pode-se então elaborar um classificador probabilístico de notícias baseado na média amostral da variável aleatória R^2.



(a) Função de Densidade de Probabilidade

(b) Função de Distribuição Acumulada

Figura 6.4: Comportamento estatístico de todo o *corpus* dividido de acordo com o tipo. É notório um certo deslocamento vertical nas quantidades por R^2 bem como nas probabilidades acumuladas dependendo da notícia.

6.5 A Avaliação dos Resultados

O processo de avaliação da qualidade da detecção de cada metodologia proposta baseia-se no cálculo das métricas acurácia, precisão e sensibilidade. A Figura 6.5 retrata os melhores resultados obtidos em cada metodologia, especificando seus desempenhos em cada métrica.

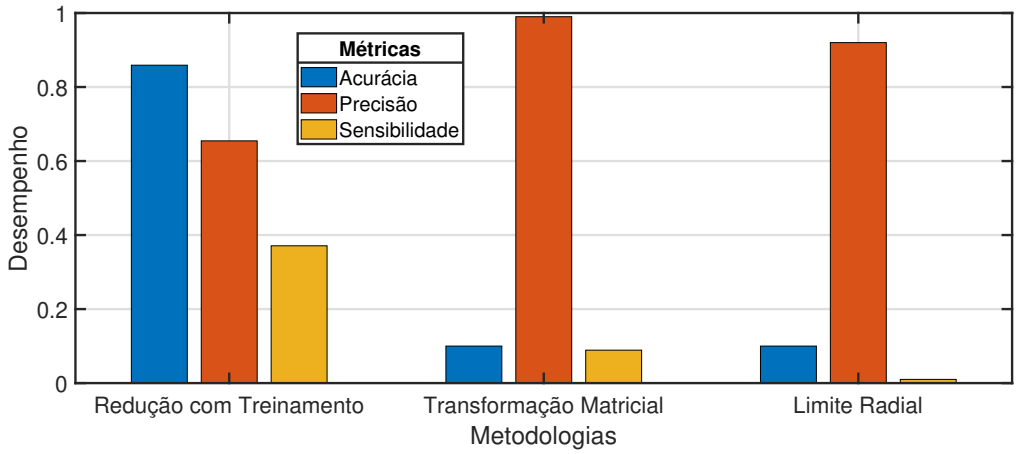


Figura 6.5: Uma comparação das metodologias revela um comportamento diverso porém ligeiramente complementar nos níveis de recuperação de informação.

Nos resultados da metodologia de redução com treinamento, a proposta demonstra um desempenho mais homogêneo entre as métricas, destacando-se principalmente pela alta acurácia e porcentagem de sensibilidade mais expressiva dentre as três metodologias. Já nos resultados referentes à metodologia de transformação matricial usando a função

de núcleo linear, percebe-se uma clara predominância na habilidade de classificar qualitativamente notícias como sendo falsas. Em contrapartida, detém baixas porcentagens de acurácia e sensibilidade, estas possivelmente fruto de perdas de características, importantes na diferenciação das notícias, impostas por dois níveis de redução dimensional – LSI e *k-means*. Analogamente à anterior, a metodologia de limite radial apresenta o mesmo caráter preciso na identificação de notícias falsas embora constate-se uma depreciação nos seus níveis de sensibilidade.

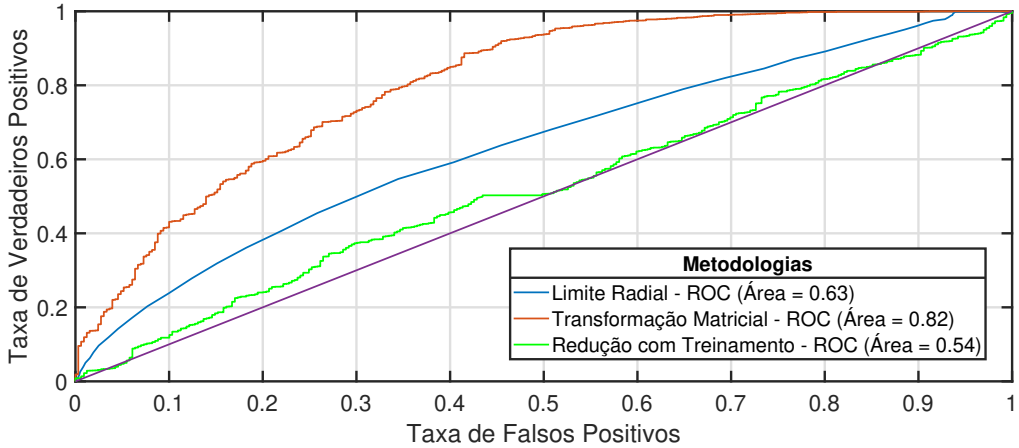


Figura 6.6: As curvas ROC refletem o desempenho de um sistema classificador binário à medida que o seu limiar de discriminação varia. Dentre as metodologias, a transformação matricial apresenta o melhor desempenho, visto que possui a maior área acima da reta.

Como uma avaliação adicional, opta-se por verificar a Curva Característica de Operação do Receptor, a ROC, de cada metodologia expressa na Figura 6.6. O bom resultado obtido pela transformação matricial,área abaixo da curva de 0,82, baseia-se no fato de que ao realizar o agrupamento com o *k-means*, a dimensionalidade dos dados é substancialmente reduzida, permitindo que a SVM de classe única defina uma hiper-superfície mais ajustada aos dados. O resultado do limite radial baseia-se no teste de hipótese realizado *a priori*, que mostra que a média do somatório dos valores quadráticos das frequências entre notícias verdadeiras e falsas são diferentes. O limite radial apresenta bom resultado pois notícias falsas tendem a ter maior frequência de uso das palavras, Figura 6.4(b), enquanto notícias verídicas tendem a variar palavras, usando mais vocabulário. Como a diferença nas distribuições é pequena, o classificador baseado nessa distribuição tem um resultado mediano, apresentado uma área abaixo da curva de 0,63. Embora esta regra possa indicar que a metodologia de transformação matricial é a escolha ótima de detecção, prefere-se compor uma decisão final ponderada em todos os resultados.

Capítulo 7

Conclusão

O cenário de iminente expansão de dados, sobretudo na forma textual, eleva o Processamento de Linguagem Natural como um campo um campo de pesquisa de bastante interesse, visto que há muitos dados a serem explorados. Sua habilidade de analisar a linguagem na sua forma mais bruta, por vezes, sem qualquer padronização, estruturação ou concordância, e mesmo assim transformá-la em representações significativas passíveis de interpretadas por computadores, torna-o adaptável à várias aplicações. Abrangendo três dessas aplicações, sumarização de textos, representação de ontológica e detecção de notícias falsas, a presente dissertação propõe três abordagens que implementam eficientemente o processamento de linguagem natural.

A primeira parte da dissertação contempla a geração automática de resumos extrativos em português, a partir de textos em diferentes formatos. Como contribuição, esse trabalho propôs a Rezzumin, uma ferramenta *web* de código aberto, que introduz uma abordagem híbrida para a ponderação entre a diversidade de tópicos e a concisão do resumo final. Além das tradicionais técnicas de processamento textual, a Rezzumin incorporou um mapeamento com tesauro garantindo assim uma otimização sem perdas significativas na semântica. A avaliação da ferramenta usando uma base de artigos científicos evidencia que a abordagem proposta apresenta quantidade de informação equiparável entre o resumo extrativo gerado e o abstrativo presente nos artigos avaliados quando ambos têm comprimentos semelhantes. Também é importante destacar um aumento significativo de até 28% no desempenho de todas as metodologias que foram enriquecidas com o tesauro.

Migrando para abordagem mais granular que privilegia a identificação de termos relevantes em frases, a segunda parte da dissertação concentrou-se na construção de estruturas ontológicas. Nesse sentido, o trabalho propôs uma abordagem de relacionamento sintático baseada no processamento de linguagem natural, que aplica diferentes algoritmos de

agrupamento na criação de uma taxonomia, a partir de textos extraídos de documentos científicos. O processo inicia-se pela coleta e análise linguística de dezenas de artigos relacionados ao domínio da computação em nuvem, em arquivos no formato PDF (*Portable Document Format*), permitindo a identificação de relações sintáticas recorrentes entre os termos. Ao transformar essas estruturas em um grafo direcionado, uma representação vetorial é obtida, tornando viável o procedimento de redução dimensional. Após uma redução de mais de um quinto do número original de características, três algoritmos de agrupamento foram implementados, o *k-medoids*, o hierárquico e o DBSCAN. O processo de validação da proposta incluiu uma avaliação bilateral. A primeira comparando as três metodologias de agrupamento entre si, usando métricas intrínsecas. Na segunda avaliação, a proposta foi comparada com outra metodologia do estado da arte. Os resultados apontaram uma superioridade em todos os cenários de agrupamento da metodologia proposta, em que se destacam a popularidade até 2,5 vezes maior dos termos identificados pela proposta e cobertura até 6 vezes maior que a metodologia concorrente.

Por fim, esta dissertação adentra os desafios envolvendo a identificação de notícias falsas, um exemplo bastante atual do estudo sobre detecção de anomalias em dados em linguagem natural. Com esta finalidade, apresenta-se uma análise estilístico-computacional, baseada em processamento de linguagem natural, que aplica eficientemente algoritmos de aprendizagem não supervisionada na detecção de notícias falsas em textos extraídos de mídias sociais. Para alcançar o objetivo, primeiramente foi construída uma base de dados composta por notícias falsas e legítimas coletadas do *Twitter*. Em seguida, aplica-se nessa base de dados técnicas de processamento de linguagem natural, incluindo a *tokenização*, remoção de termos de menor significância, correção ortográfica e *stemização*. Uma vez padronizadas, cada sentença foi exposta ao modelo de vetorização TF-IDF, garantindo que as sentenças se tornassem matematicamente operáveis. Posteriormente, visando reduzir a dimensionalidade opta-se por aplicar a técnica de indexação semântica latente (LSI). Após uma redução de mais de 85% no número de características, foram implementadas três metodologias de classificação de notícias distintas. No processo de avaliação da qualidade da detecção das metodologias destaca-se uma acurácia de 86% e precisão de 94%. Como as metodologias propostas apresentam resultados complementares, conclui-se que uma boa solução para problema de detecção de notícias falsas seria uma abordagem que mesclasse pelo menos duas dessas metodologias em série.

Esta dissertação compila o conteúdo de um conjunto de trabalhos publicados em congresso (*International Conference on Computing, Networking and Communications* - ICNC 2020) e em revista (*IEEE Signal Processing Letters* - IEEE SPL), bem como trabalhos

aguardando publicação em minicurso (Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais - SBSeg 2020) e em congresso (*Cloud and Internet of Things* - CIoT 2020).

Como trabalhos futuros, primeiramente planeja-se incorporar à ferramenta Rezzumin novas métricas de ponderação vetorial, a funcionalidade de detecção de idioma, além de adicionar uma avaliação sobre grau de redundância dos resumos gerados. Em relação a abordagem de construção de ontologias, pretende-se desenvolver a tarefa de descoberta de axiomas. Com o intuito de aperfeiçoar a abordagem de detecção de notícias falsas, deseja-se ampliar a base de dados utilizada, buscando outras fontes de notícias falsas, como agências de verificação manual de notícias. Ademais, visa-se acrescentar algoritmos de agrupamento não supervisionado, discriminando assim as notícias por temas, como política, entretenimento e economia.

Referências

[1] CASAGRANDE, E.; ARNAUTOVIC, E.; WOON, W. L.; ZEINELDIN, H. H.; SVETINOVIC, D. Semiautomatic system domain data analysis: a smart grid feasibility case study. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, IEEE, v. 47, n. 12, p. 3117–3127, 2016.

[2] GAMBHIR, M.; GUPTA, V. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, v. 47, n. 1, p. 1–66, 2017.

[3] GUPTA, V.; LEHAL, G. S. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, Citeseer, v. 2, n. 3, p. 258–268, 2010.

[4] AL-ASWADI, F. N.; CHAN, H. Y.; GAN, K. H. Automatic ontology construction from text: a review from shallow to deep learning trend. *Artificial Intelligence Review*, Springer, p. 1–28, 2019.

[5] WONG, W.; LIU, W.; BENNAMOUN, M. Ontology learning from text: A look back and into the future. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 44, n. 4, p. 1–36, 2012.

[6] SERRA, I.; GIRARDI, R. Extracting non-taxonomic relationships of ontologies from texts. In: SPRINGER. *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*. [S.l.], 2011. p. 329–338.

[7] ASIM, M. N.; WASIM, M.; KHAN, M. U. G.; MAHMOOD, W.; ABBASI, H. M. A survey of ontology learning techniques and applications. *Database*, Oxford University Press, v. 2018, 2018.

[8] BROWARNIK, A.; MAIMON, O. Ontology learning from text: why the ontology learning layer cake is not viable. *International Journal of Signs and Semiotic Systems (IJSSS)*, IGI Global, v. 4, n. 2, p. 1–14, 2015.

[9] ALBUKHITAN, S.; HELMY, T.; ALNAZER, A. Arabic ontology learning using deep learning. In: *Proceedings of the International Conference on Web Intelligence*. New York, NY, USA: Association for Computing Machinery, 2017. (WI '17), p. 1138–1142.

[10] LIU, G.; WANG, Y.; ORGUN, M. A. Quality of trust for social trust path selection in complex social networks. In: INTERNATIONAL FOUNDATION FOR AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS. *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. [S.l.], 2010. p. 1575–1576.

[11] GABIELKOV, M.; RAMACHANDRAN, A.; CHAINTREAU, A.; LEGOUT, A. Social Clicks: What and Who Gets Read on Twitter? In: *ACM SIGMETRICS / IFIP Performance 2016*. Antibes Juan-les-Pins, France: [s.n.], 2016. Disponível em: <https://hal.inria.fr/hal-01281190>.

[12] ZHOU, X.; ZAFARANI, R. Fake news: A survey of research, detection methods, and opportunities. *arXiv preprint arXiv:1812.00315*, 2018.

[13] OLIVEIRA, N. R. de; REIS, L. H.; FERNANDES, N. C.; BASTOS, C. A. M.; MEDEIROS, D. S. V. de; MATTOS, D. M. F. Natural language processing characterization of recurring calls in public security services. In: IEEE. *Proceedings of the 2020 International Conference on Computing, Networking and Communications (ICNC)*. [S.l.], 2020. p. 1009–1013.

[14] CLARK, M.; KIM, Y.; KRUSCHWITZ, U.; SONG, D.; ALBAKOUR, D.; DIGNUM, S.; BERESI, U. C.; FASLI, M.; ROECK, A. D. Automatically structuring domain knowledge from text: An overview of current research. *Information Processing & Management*, Elsevier, v. 48, n. 3, p. 552–568, 2012.

[15] OTTER, D. W.; MEDINA, J. R.; KALITA, J. K. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, 2020.

[16] BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2009. ISBN 0596516495, 9780596516499.

[17] NAVIGLI, R. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 41, n. 2, p. 1–69, 2009.

[18] MANNING, C.; SURDEANU, M.; BAUER, J.; FINKEL, J.; BETHARD, S.; MCCLOSKY, D. The stanford corenlp natural language processing toolkit. In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. [S.l.: s.n.], 2014. p. 55–60.

[19] MANNING, C.; RAGHAVAN, P.; SCHÜTZE, H. Introduction to information retrieval. *Natural Language Engineering*, Cambridge university press, v. 16, n. 1, p. 100–103, 2010.

[20] ZHAI, Y.; ONG, Y.-S.; TSANG, I. W. The emerging"big dimensionality". *IEEE Computational Intelligence Magazine*, IEEE, v. 9, n. 3, p. 14–26, 2014.

[21] PAPADIMITRIOU, C. H.; RAGHAVAN, P.; TAMAKI, H.; VEMPALA, S. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, Elsevier, v. 61, n. 2, p. 217–235, 2000.

[22] DEERWESTER, S.; DUMAIS, S. T.; FURNAS, G. W.; LANDAUER, T. K.; HARSHMAN, R. Indexing by latent semantic analysis. *Journal of the American society for information science*, Wiley Online Library, v. 41, n. 6, p. 391–407, 1990.

[23] MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-Hill, 1997. (McGraw Hill series in computer science).

[24] VERIKAS, A.; GELZINIS, A.; BACAUSKIENE, M. Mining data with random forests: A survey and results of new tests. *Pattern recognition*, Elsevier, v. 44, n. 2, p. 330–349, 2011.

[25] KADHIM, A. I. Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, Springer, v. 52, n. 1, p. 273–292, 2019.

[26] XU, R.; WUNSCH, D. Survey of clustering algorithms. *IEEE Transactions on neural networks*, Ieee, v. 16, n. 3, p. 645–678, 2005.

[27] FAHAD, A.; ALSHATRI, N.; TARI, Z.; ALAMRI, A.; KHALIL, I.; ZOMAYA, A. Y.; FOUFOU, S.; BOURAS, A. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, IEEE, v. 2, n. 3, p. 267–279, 2014.

[28] KETCHEN, D. J.; SHOOK, C. L. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, Wiley Online Library, v. 17, n. 6, p. 441–458, 1996.

[29] ROUSSEEUW, P. J.; KAUFMAN, L. Finding groups in data. *Hoboken: Wiley Online Library*, Wiley Online Library, 1990.

[30] GAN, J.; TAO, Y. Dbscan revisited: Mis-claim, un-fixability, and approximation. In: *Proceedings of the 2015 ACM SIGMOD international conference on management of data*. [S.l.: s.n.], 2015. p. 519–530.

[31] SCHUBERT, E.; SANDER, J.; ESTER, M.; KRIEGEL, H. P.; XU, X. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, ACM New York, NY, USA, v. 42, n. 3, p. 1–21, 2017.

[32] BENAVENT, X.; CASTELLANOS, A.; VES, E. de; GARCIA-SERRANO, A.; CIGARRAN, J. Fca-based knowledge representation and local generalized linear models to address relevance and diversity in diverse social images. *Future Generation Computer Systems*, Elsevier, v. 100, p. 250–265, 2019.

[33] CURISKIS, S. A.; DRAKE, B.; OSBORN, T. R.; KENNEDY, P. J. An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit. *Information Processing & Management*, Elsevier, v. 57, n. 2, p. 102034, 2020.

[34] XIAO, J.; LU, J.; LI, X. Davies bouldin index based hierarchical initialization k-means. *Intelligent Data Analysis*, IOS Press, v. 21, n. 6, p. 1327–1338, 2017.

[35] VERGANI, A. A.; BINAGHI, E. A soft davies-bouldin separation measure. In: IEEE. *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.], 2018. p. 1–8.

[36] LIU, Y.; LI, Z.; XIONG, H.; GAO, X.; WU, J. Understanding of internal clustering validation measures. In: IEEE. *2010 IEEE International Conference on Data Mining*. [S.l.], 2010. p. 911–916.

[37] BANKO, M.; BRILL, E. Scaling to very very large corpora for natural language disambiguation. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 39th annual meeting on association for computational linguistics*. [S.l.], 2001. p. 26–33.

[38] SOCHER, R.; PERELYGIN, A.; WU, J.; CHUANG, J.; MANNING, C. D.; NG, A.; POTTS, C. Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. [S.l.: s.n.], 2013. p. 1631–1642.

[39] RUNESON, P.; ALEXANDERSSON, M.; NYHOLM, O. Detection of duplicate defect reports using natural language processing. In: IEEE COMPUTER SOCIETY. *Proceedings of the 29th international conference on Software Engineering*. [S.l.], 2007. p. 499–510.

[40] CARDOSO, P. C.; MAZIERO, E. G.; JORGE, M. L.; SENO, E. M.; FELIPPO, A. D.; RINO, L. H.; NUNES, M. d. G. V.; PARDO, T. A. CSTnews-a discourse-annotated corpus for single and multi-document summarization of news texts in brazilian portuguese. In: *Proceedings of the 3rd RST Brazilian Meeting*. [S.l.: s.n.], 2011. p. 88–105.

[41] CARDOSO, P. C.; PARDO, T. A. Multi-document summarization using semantic discourse models. *Procesamiento del Lenguaje Natural*, Sociedad Española para el Procesamiento del Lenguaje Natural, n. 56, p. 57–64, 2016.

[42] OLIVEIRA, H.; FERREIRA, R.; LIMA, R.; LINS, R. D.; FREITAS, F.; RISS, M.; SIMSKE, S. J. Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. *Expert Systems with Applications*, Elsevier, v. 65, p. 68–86, 2016.

[43] RANI, M.; DHAR, A. K.; VYAS, O. Semi-automatic terminology ontology learning based on topic modeling. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 63, p. 108–125, 2017.

[44] PETRUCCI, G.; ROSPOCHER, M.; GHIDINI, C. Expressive ontology learning as neural machine translation. *Journal of Web Semantics*, Elsevier, v. 52, p. 66–82, 2018.

[45] KAIYA, H.; SAEKI, M. Using domain ontology as domain knowledge for requirements elicitation. In: IEEE. *14th IEEE International Requirements Engineering Conference (RE'06)*. [S.l.], 2006. p. 189–198.

[46] BABBAR, R.; PARTALAS, I.; GAUSSIER, E.; AMINI, M.-R.; AMBLARD, C. Learning taxonomy adaptation in large-scale classification. *The Journal of Machine Learning Research*, JMLR. org, v. 17, n. 1, p. 3350–3386, 2016.

[47] LIU, N.; HUANG, X.; LI, J.; HU, X. On interpretation of network embedding via taxonomy induction. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. [S.l.: s.n.], 2018. p. 1812–1820.

[48] ZOUAQ, A.; MARTEL, F. What is the schema of your knowledge graph? leveraging knowledge graph embeddings and clustering for expressive taxonomy learning. In: *Proceedings of The International Workshop on Semantic Big Data*. [S.l.: s.n.], 2020. p. 1–6.

[49] WOON, W. L.; MADNICK, S. Asymmetric information distances for automated taxonomy construction. *Knowledge and information systems*, Springer, v. 21, n. 1, p. 91–111, 2009.

[50] RASHKIN, H.; CHOI, E.; JANG, J. Y.; VOLKOVA, S.; CHOI, Y. Truth of varying shades: Analyzing language in fake news and political fact-checking. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. [S.l.: s.n.], 2017. p. 2931–2937.

[51] PENNEBAKER, J. W.; FRANCIS, M. E.; BOOTH, R. J. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, v. 71, n. 2001, p. 2001, 2001.

[52] RUBIN, V.; CONROY, N.; CHEN, Y.; CORNWELL, S. Fake news or truth? using satirical cues to detect potentially misleading news. In: *Proceedings of the second workshop on computational approaches to deception detection*. [S.l.: s.n.], 2016. p. 7–17.

[53] CHEN, Y.; CONROY, N. J.; RUBIN, V. L. Misleading online content: Recognizing clickbait as false news. In: ACM. *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*. [S.l.], 2015. p. 15–19.

[54] FERREIRA, W.; VLACHOS, A. Emergent: a novel data-set for stance classification. In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*. [S.l.: s.n.], 2016. p. 1163–1168.

[55] WANG, W. Y. “Liar, liar pants on fire": A new benchmark dataset for fake news detection. In: *Annual Meeting of the Association for Computational Linguistics - ACL 2017*. [S.l.: s.n.], 2017.

[56] MONTEIRO, R. A.; SANTOS, R. L.; PARDO, T. A.; ALMEIDA, T. A. de; 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.

[57] BARRETO, H. F.; CAMPISTA, M. E. M.; COSTA, L. H. M. Spammers no twitter: Quando contatos deixam de ser bem-vindos. In: *Workshop de Redes P2P, Dinâmicas, Sociais e Orientadas a Conteúdo (Wp2p+ 2014) - SBRC 2014*. [S.l.: s.n.], 2014. v. 1, p. 23–36.

[58] ALLAHYARI, M.; POURIYEH, S.; ASSEFI, M.; SAFAEI, S.; TRIPPE, E. D.; GUTIERREZ, J. B.; KOCHUT, K. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*, 2017.

[59] Amato, F.; Santo, A. D.; Moscato, V.; Picariello, A.; Serpico, D.; Sperlì, G. A lexicon-grammar based methodology for ontology population for e-health applications. In: *2015 Ninth International Conference on Complex, Intelligent, and Software Intensive Systems*. [S.l.: s.n.], 2015. p. 521–526.

[60] CHETTRI, R.; CHAKRABORTY, U. K. Automatic text summarization. *International Journal of Computer Applications*, Foundation of Computer Science, v. 161, n. 1, p. 5–7, 2017.

[61] Jafari, M.; Wang, J.; Qin, Y.; Gheisari, M.; Shahabi, A. S.; Tao, X. Automatic text summarization using fuzzy inference. In: *2016 22nd International Conference on Automation and Computing (ICAC)*. [S.l.: s.n.], 2016. p. 256–260.

[62] Singh, S. P.; Kumar, A.; Mangal, A.; Singhal, S. Bilingual automatic text summarization using unsupervised deep learning. In: *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. [S.l.: s.n.], 2016. p. 1195–1200.

[63] LYNN, H. M.; CHOI, C.; KIM, P. An improved method of automatic text summarization for web contents using lexical chain with semantic-related terms. *Soft Computing*, Springer, v. 22, n. 12, p. 4013–4023, 2018.

[64] LOPEZ, M. A.; MATTOS, D. M. F.; DUARTE, O. C. M. B. An elastic intrusion detection system for software networks. *Annals of Telecommunications*, v. 71, n. 11, p. 595–605, 2016.

[65] GRUBER, T. R. A translation approach to portable ontology specifications. *Knowledge acquisition*, Academic Press, v. 5, n. 2, p. 199–220, 1993.

[66] BUITELAAR, P.; CIMIANO, P.; MAGNINI, B. *Ontology learning from text: methods, evaluation and applications*. [S.l.]: IOS press, 2005.

[67] NAVIGLI, R.; VELARDI, P. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, MIT Press, v. 30, n. 2, p. 151–179, 2004.

[68] MCDANIEL, M.; STOREY, V. C. Evaluating domain ontologies: clarification, classification, and challenges. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 52, n. 4, p. 1–44, 2019.

[69] OLIVEIRA, N. R. de; MEDEIROS, D. S. V. de; MATTOS, D. M. F. Syntactic-relationship approach to construct well-informative knowledge graphs representation. In: IEEE. *4th Cloud and Internet of Things - CIoT'20 (a ser apresentado)*. [S.l.], 2020.

[70] CILIBRASI, R. L.; VITANYI, P. M. The google similarity distance. *IEEE Transactions on knowledge and data engineering*, IEEE, v. 19, n. 3, p. 370–383, 2007.

[71] HEYMANN, P.; GARCIA-MOLINA, H. *Collaborative creation of communal hierarchical taxonomies in social tagging systems*. [S.l.], 2006.

[72] VOSOUGHI, S.; ROY, D.; ARAL, S. The spread of true and false news online. *Science*, American Association for the Advancement of Science, v. 359, n. 6380, p. 1146–1151, 2018. ISSN 0036-8075. Disponível em: <https://science.sciencemag.org/content/359/6380/1146>.

[73] ISLAM, M. S.; SARKAR, T.; KHAN, S. H.; KAMAL, A.-H. M.; HASAN, S. M. M.; KABIR, A.; YEASMIN, D.; ISLAM, M. A.; CHOWDHURY, K. I. A.; ANWAR, K. S. Covid-19–related infodemic and its impact on public health: A global social media analysis. *The American Journal of Tropical Medicine and Hygiene*, ASTMH, 2020.

[74] RUBIN, V. L. On deception and deception detection: Content analysis of computer-mediated stated beliefs. In: AMERICAN SOCIETY FOR INFORMATION SCIENCE. *Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem-Volume 47*. [S.l.], 2010. p. 32.

[75] RUBIN, V. L.; CHEN, Y.; CONROY, N. J. Deception detection for news: three types of fakes. In: AMERICAN SOCIETY FOR INFORMATION SCIENCE. *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*. [S.l.], 2015. p. 83.

[76] SHU, K.; SLIVA, A.; WANG, S.; TANG, J.; LIU, H. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, ACM, v. 19, n. 1, p. 22–36, 2017.

[77] ZUCKERMAN, M.; DEPAULO, B. M.; ROSENTHAL, R. Verbal and nonverbal communication of deception. In: *Advances in experimental social psychology*. [S.l.]: Elsevier, 1981. v. 14, p. 1–59.

[78] NYHAN, B.; REIFLER, J. When corrections fail: The persistence of political misperceptions. *Political Behavior*, Springer, v. 32, n. 2, p. 303–330, 2010.

[79] PRONIN, E.; KRUGER, J.; SAVTISKY, K.; ROSS, L. You don't know me, but i know you: The illusion of asymmetric insight. *Journal of Personality and Social Psychology*, American Psychological Association, v. 81, n. 4, p. 639, 2001.

[80] de Oliveira, N. R.; Medeiros, D. S. V.; Mattos, D. M. F. A sensitive stylistic approach to identify fake news on social networking. *IEEE Signal Processing Letters*, v. 27, p. 1250–1254, 2020.

[81] OSHIKAWA, R.; QIAN, J.; WANG, W. Y. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*, 2018.

[82] THORNE, J.; VLACHOS, A.; CHRISTODOULOPOULOS, C.; MITTAL, A. FEVER: a large-scale dataset for fact extraction and VERification. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. [S.l.]: Association for Computational Linguistics, 2018. p. 809–819.

[83] SANTIA, G. C.; WILLIAMS, J. R. Buzzface: A news veracity dataset with facebook user commentary and egos. In: *Twelfth International AAAI Conference on Web and Social Media*. [S.l.: s.n.], 2018. p. 531–540.

[84] ZUBIAGA, A.; LIAKATA, M.; PROCTER, R.; HOI, G. W. S.; TOLMIE, P. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, Public Library of Science San Francisco, CA USA, v. 11, n. 3, p. e0150989, 2016.

[85] MITRA, T.; GILBERT, E. Credbank: A large-scale social media corpus with associated credibility annotations. In: *ICWSM*. [S.l.: s.n.], 2015. p. 258–267.

[86] SHU, K.; MAHUDESWARAN, D.; WANG, S.; LEE, D.; LIU, H. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, Mary Ann Liebert, Inc., publishers 140 Huguenot Street, 3rd Floor New ..., v. 8, n. 3, p. 171–188, 2020.

[87] RUBIN, V. L. Pragmatic and cultural considerations for deception detection in asian languages. 2014.

[88] Perdisci, R.; Gu, G.; Lee, W. Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems. In: *Sixth International Conference on Data Mining (ICDM'06)*. [S.l.: s.n.], 2006. p. 488–498. ISSN 2374-8486.

[89] Gaonkar, S.; Itagi, S.; Chalippatt, R.; Gaonkar, A.; Aswale, S.; Shetgaonkar, P. Detection of online fake news : A survey. In: *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*. [S.l.: s.n.], 2019. p. 1–6.