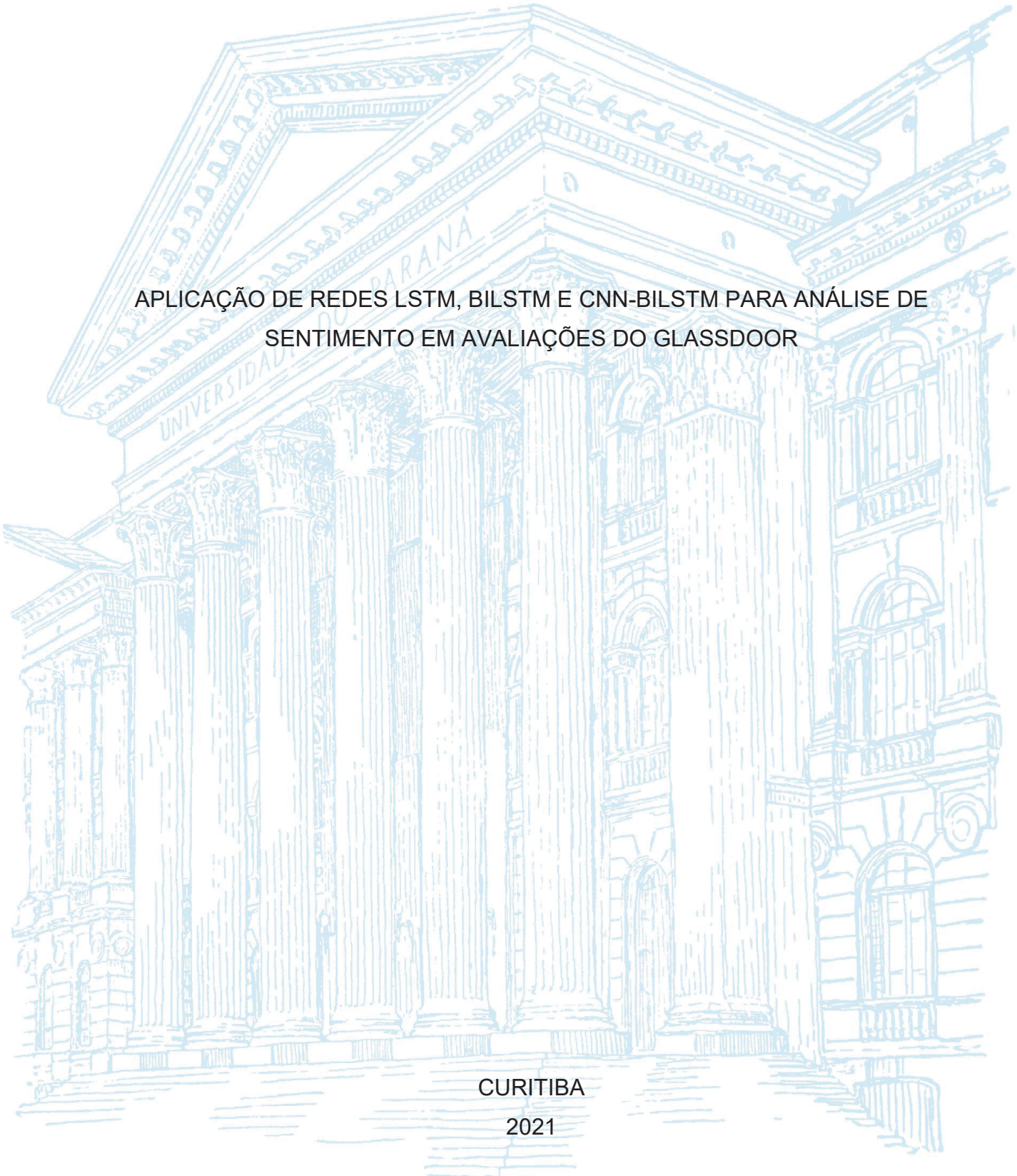
UNIVERSIDADE FEDERAL DO PARANÁ

WESLEY MAFFAZZOLLI



APLICAÇÃO DE REDES LSTM, BILSTM E CNN-BILSTM PARA ANÁLISE DE SENTIMENTO EM AVALIAÇÕES DO GLASSDOOR

CURITIBA

2021

WESLEY MAFFAZZOLLI

APLICAÇÃO DE REDES LSTM, BILSTM E CNN-BILSTM PARA ANÁLISE DE SENTIMENTO EM AVALIAÇÕES DO GLASSDOOR

Monografia apresentada ao curso de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada.

Orientador: Prof. Dr. Razer Anthom Nizer Rojas Montaño

CURITIBA

2021

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **WESLEY MAFFAZZOLLI** intitulada: **Aplicação de Redes LSTM, BiLSTM e CNN-BiLSTM para Análise de Sentimento em Avaliações do Glassdoor**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua _APROVAÇÃO_____ no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 11 de Novembro de 2021.

RAZER ANTHOM NIZER ROJAS MONTAÑO
Presidente da Banca Examinadora

ALEXANDER ROBERT KUTZKE
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Aplicação de Redes LSTM, BiLSTM e CNN-BiLSTM para Análise de Sentimento em Avaliações do Glassdoor

Wesley Maffazzolli
Especialização em Inteligência Artificial Aplicada
Universidade Federal do Paraná (UFPR)
Curitiba, Brasil
wesley.maffazzolli@gmail.com.br

Razer Anthom Nizer Rojas Montaño
Especialização em Inteligência Artificial Aplicada
Universidade Federal do Paraná (UFPR)
Curitiba, Brasil
razer@ufpr.br

***Resumo*— A pandemia de Covid-19 impôs a transformação digital como um quesito de sobrevivência para as empresas. Neste sentido, o mercado já assolado pela escassez de profissionais de TI, enfrenta uma competição sem precedentes para conquistar especialistas na área, cada vez mais exigentes e valiosos. Para concretizar suas estratégias digitais, as empresas precisam adotar práticas eficazes para atrair e reter talentos, como as baseadas na escuta ativa de profissionais em canais internos e externos. Neste sentido, este artigo tem como objetivo investigar a potencialidade da aplicação de modelos baseados em redes LSTM, BiLSTM e CNN-BiLSTM para a análise de sentimento de avaliações textuais publicadas no site de vagas e recrutamento Glassdoor. Desta forma, espera-se auxiliar empresas no acompanhamento de percepções publicadas por profissionais em relação às suas marcas a fim de aprimorarem suas estratégias de atração e retenção de talentos.**

***Palavras-chave:* classificação, análise, sentimento, LSTM, BiLSTM, CNN-BiLSTM, avaliações, Glassdoor**

***Abstract*—The Covid-19 pandemic imposed digital transformation as a matter of survival for companies. In this sense, the market already plagued by the shortage of IT professionals, faces an unprecedented competition to win specialists in the area, which are increasingly demanding and valuable. To realize their digital strategies, companies need to adopt effective practices to attract and retain talent, such as those based on active listening to professionals in internal and external channels. In this sense, this article aims to investigate the potential of applying models based on LSTM, BiLSTM and CNN-BiLSTM networks for the analysis of the sentiment of textual evaluations published on the vacancies and recruitment website Glassdoor. In this way, it is expected to help companies in monitoring the perceptions published by professionals in relation to their brands in order to improve their talent attraction and retention strategies.**

***Index Terms*—classification, sentiment, analysis, LSTM, BiLSTM, CNN-BiLSTM, evaluation, Glassdoor**

I. Introdução

A pandemia de Covid-19 afetou profundamente a economia do planeta, exigindo uma ação direta das esferas pública e privada para o enfrentamento da situação e minimização dos seus impactos (DELOITTE, 2020). Diante das restrições sanitárias e de distanciamento social impostas pela pandemia de Covid-19 (OMS, 2021), atividades essenciais como estudar, trabalhar, consumir e socializar foram limitadas. Neste sentido, as pessoas passaram a adquirir novos hábitos para enfrentar as restrições, sendo as tecnologias essenciais para estas adaptações (GLOBO, 2020).

Segundo o Sebrae (2020), a pandemia foi responsável por incorporar o digital no cotidiano da sociedade e dos negócios. Para as pessoas, por exemplo, uma importante mudança na rotina foi a adoção do trabalho e estudo remoto por ferramentas de videoconferência (GLOBO, 2020). Paralelamente, as empresas também tiveram que se adaptar e acelerar seus processos de digitalização (SEBRAE, 2020). Na prática, conforme conclui a pesquisa conduzida pelo *The Economist Intelligence Unit* com líderes de negócios ao redor mundo, organizações digitalmente preparadas conseguiram se adaptar, guiar e crescer durante a pandemia (UNIT, 2021). Desta forma, a transformação digital deixou de ser opcional e tornou-se crítica para a sobrevivência das empresas (GOASDUFF, 2020).

Neste sentido, para auxiliar o mercado neste processo de transformação, é essencial que o setor de TI acompanhe a movimentação da demanda por profissionais de TI que, segundo Sena e Granato (2021), é um reflexo global. Afinal, o profissional de tecnologia é fundamental para conduzir o processo de transformação digital (G1, 2021) e promover o progresso das empresas (GPTW, 2020).

Diante da importância destes profissionais para a concretização das estratégias digitais e a alta competitividade do mercado, as empresas necessitam se tornar atraentes para contratá-los (SENA; GRANATO, 2021) e retê-los (ROSA; RIBAS, 2021). Para isso, elas podem adotar estratégias de recrutamento e retenção, como a construção contínua de uma marca empregadora *Employer Branding* (EB). Através dessa marca, as organizações comunicam a proposta de valor que oferecem para seus colaboradores (público interno) e profissionais em prospecção (público externo), conhecida como *Employee Value Proposition* (EVP). Na prática, a EVP funciona como a base da marca empregadora, a qual representa aquilo que a empresa oferece ao encontro daquilo que os profissionais valorizam (ARMSTRONG; TAYLOR, 2014).

Entretanto, para alcançar esta relação, a organização necessita, primeiramente, identificar quais são estes pontos de

valorização (ARMSTRONG; TAYLOR, 2014). Neste sentido, uma das formas de realizar isto é por meio da escuta interna da empresa, como a coleta de dados em pesquisas de clima organizacional (GPTW, 2020). Alternativamente, essa escuta também pode ser realizada de forma externa, a partir da análise dos dados de publicações encaminhadas por atuais e ex-colaboradores em sites de recrutamento e avaliação de empresas (CONNLEY, 2021), como o Glassdoor.

Em geral, estes processos de escuta demandam esforço de análise contínuos para a extração de conhecimentos sobre as percepções da organização. É no contexto destas atividades que a aplicação de modelos inteligentes podem auxiliar empresas a acompanharem continuamente as avaliações da sua marca. Desta forma, este artigo visa explorar oportunidades de aplicações de modelos, no contexto da escuta organizacional, para a classificação de sentimentos em avaliações textuais publicadas por colaboradores. Além disso, utilizando como base para criação destes modelos publicações disponíveis no portal Glassdoor. Com isso, espera-se que as empresas possam adquirir mais eficiência no acompanhamento e avaliações das opiniões de seus colaboradores, a fim de ampliar seu conhecimento sobre sua EVP e, deste modo, melhorar seus índices de contratação e retenção de profissionais.

A. Motivações

A definição do tema e propósito para a criação deste estudo foi baseada em dois motivadores. O primeiro, partiu da criação de um modelo de análise de sentimento que pudesse auxiliar empresas no acompanhamento do sentimento dos seus colaboradores. Com isso, tal modelo de aprendizado poderia contribuir para o ganho de eficiência em análises de textos publicados por colaboradores em canais internos e externos da organização e, desta forma, auxiliar na melhoria de indicadores de contratação e retenção de profissionais.

Como base para o primeiro motivador, cabe salientar que o Brasil vive um déficit de profissionais no mercado que é também um fator histórico. Conforme aponta o relatório anual da Brasscom, isto ocorre porque o Brasil gradua apenas 46 mil dos 70 mil profissionais de TI demandados por ano, totalizando um déficit de 260 mil profissionais até 2024 (BRASSCOM, 2021). Na pandemia, esta escassez ficou ainda mais evidente devido ao receio das empresas de ficar sem estes profissionais, o que causaria possíveis paradas em suas iniciativas digitais, que são essenciais para a sua sobrevivência. Mesmo pagando maiores salários para conquistá-los (SENA; GRANATO, 2021), os índices de rotatividade também são altos. Afinal, os profissionais ainda são constantemente assediados por outras empresas, até mesmo, do exterior, que conseguem atrair profissionais com salários mais altos e que dispensam burocracias de emigração por conta do *home-office* (ROSA; RIBAS, 2021). Embora a remuneração seja um fator relevante, para estes profissionais o desenvolvimento da carreira, equilíbrio entre vida pessoal e profissional, assim como a cultura e o ambiente da empresa também são determinantes (INDEED, 2020).

Do outro lado, as empresas buscam estratégias para superar os desafios deste cenário e melhorar seus índices de contratação e retenção de talentos. Por meio da criação de uma marca empregadora, baseada numa proposta sólida de valor para seus profissionais, muitas organizações buscam ouvir seus colaboradores para alcançarem o sucesso das suas estratégias (GPTW, 2020). Desta forma, por meio de ferramentas de análise e extração de conhecimento a partir de avaliações dos profissionais, as empresas podem focar seu tempo na interpretação e descoberta dos desejos do seu público. Ao invés de se debruçarem em tarefas manuais e repetitivas para geração destes conhecimentos. Neste sentido, modelos inteligentes podem ser aplicados para realizarem estas tarefas operacionais de exploração, análise e extração de conhecimento a fim de disponibilizar insumos valiosos para os tomadores de decisões nas organizações.

O segundo motivador surgiu do contexto de uma multinacional brasileira do setor de TI que, para fins de simplificação, foi denominada "ETI" ao longo do estudo, que não possuía um modelo específico para análise de sentimento de avaliações de colaboradores. Entretanto, esta empresa já detinha um modelo baseado em redes neurais convolucionais (CNNs) utilizado para análise de sentimento de avaliações em lojas de aplicativos de seus clientes, já em produção. Aqui nomeado de "ANP", este modelo foi treinado pela ETI a partir de textos de avaliação de produtos publicados em lojas de *e-commerce*. Desta forma, o cerne do segundo motivador do estudo está no fato de que a ETI estava propensa a utilizar o ANP em avaliações publicadas por seus atuais e ex-colaboradores no site Glassdoor, mas suspeitava da possível baixa acurácia do modelo ao avaliar textos de um contexto diferente do utilizado no seu treinamento.

B. Objetivos

O objetivo geral deste artigo consiste em treinar modelos de aprendizado de máquina baseados em redes neurais recorrentes (RNRs) dos tipos LSTM, BiLSTM e CNN-BiLSTM para identificar as suas potencialidades de análise de sentimento em avaliações publicadas por usuários no portal Glassdoor. Com isto, espera-se prover um recurso inteligente que auxilie empresas na construção e manutenção da sua EP e, por consequência, contribua para melhoria dos seus indicadores de contratação e retenção de profissionais.

Neste contexto, este artigo propõe três objetivos específicos a serem verificados, sendo eles:

1) Confirmar a baixa acurácia do modelo ANP (abaixo de 0,50) ao ser submetido a predições com o dataset Glassdoor;
2) Treinar três redes neurais recorrentes distintas a fim de superar a acurácia do modelo ANP;
3) Alcançar a acurácia de pelo menos 0,70 em um dos modelos de RNRs treinados durante o estudo.

Para atender o objetivo específico 1, o modelo ANP será re-treinado a partir do $dataset$ e dos códigos fontes originais do modelo ANP providos pela ETI. Além disso, o modelo

ANP será submetido a predições com o $dataset$ Glassdoor para identificar sua acurácia.

Em relação ao objetivo específico 2, em um primeiro momento será gerado o $dataset$ base para os os treinamentos e testes dos modelos a partir da extração de avaliações do site Glassdoor, referente às páginas individuais de cada uma das 105 Melhores empresas de TI para se trabalhar no Brasil em 2020. A partir disso, serão treinadas três RNRs com arquiteturas distintas (LSTM, BiLSTM e CNN-BiLSTM). Finalmente, os modelos serão submetidos à etapa de testes para obter suas acurácias, que serão comparadas entre si e com o modelo ANP.

Por fim, para atender o objetivo específico 3, serão utilizadas como base para a criação dos modelos RNRs três arquiteturas que obtiveram resultados satisfatórios em trabalhos relacionados com o problema de análise de sentimentos. Além disso, será utilizada a técnica de *grid search* para obter a melhor acurácia a partir das combinações de hiperparâmetros sugeridos nos estudos base das arquiteturas propostas e outros trabalhos relacionados com a temática.

C. Justificativa

A escolha da análise de dados a partir de um canal digital externo, como o Glassdoor, parte dos seguintes argumentos:

- Além de se preocupar com a autenticidade da sua marca empregadora no cotidiano da organização, as empresas também devem se atentar à sua reputação *online*. Em tempos onde a transparência é um imperativo e, colaboradores e ex-colaboradores podem compartilhar online suas experiências organizacionais, candidatos podem facilmente buscar e avaliar a autenticidade da marca empregadora. Segundo uma pesquisa realizada para avaliar o impacto da marca empregadora nas organizações, 81% dos entrevistados revelaram que sentem que a experiência de trabalho promovida pelo empregador não corresponde à realidade. Para tratar este gap, as organizações devem criar uma marca em que os colaboradores reconheçam, confiem e vivam no dia-a-dia da organização (SHANDWICK; RESEARCH, 2017);
- Diante da publicação destas percepções, é fundamental que as organizações monitorem constantemente estes sites, afinal, estes dados refletem diretamente a reputação da sua marca empregadora e partem do público mais confiável: os próprios profissionais. Neste sentido, uma pesquisa revela que 84% dos profissionais em busca de um emprego mudariam para uma empresa com uma boa reputação, enquanto 69% afirmaram que, mesmo desempregados, não aceitariam uma oferta de uma empresa com má reputação (GLASSDOOR, 2014);
- Conforme afirma Gptw (2020), ouvir os colaboradores é essencial para identificar pontos de aperfeiçoamento e priorizar ações consistentes para a construção da reputação da marca empregadora. Neste sentido, além de pesquisas de clima organizacional (GPTW, 2020), prestar atenção no que é publicado em sites, como Glassdoor, é também uma forma ativa de ouvir a organização (SHANDWICK; RESEARCH, 2017);
- A escolha específica da plataforma Glassdoor deu-se por algumas razões, sendo elas: o site é o principal canal de escuta externa da empresa ETI; a disponibilidade pública de dados sobre avaliações de empresas; alto número de avaliações publicadas neste site; a relevância do site nos *rankings* de pesquisa do *Google*; a menção cotidiana do Glassdoor nos círculos sociais de profissionais de TI; a menção do site no estudo de Shandwick e Research (2017).

Já em relação à escolha de RNRs para análise de sentimento, destacam-se os seguintes pontos:

- A ampla utilização de modelos baseados em RNRs e, mais especificamente arquiteturas envolvendo LSTM's (MINAEE et al., 2021), (ZHANG, 2021), (ARORA; KHODAK; SAUNSHI, 2018), (RAO; SPASOJEVIC, 2016), (LIU et al., 2016); BiLSTM's (SACHAN; ZAHEER; SALAKHUTDINOV, 2020), (LIU; GUO, 2019), (ZHOU et al., 2016); e modelos híbridos CNN-BiLSTM (MINAEE et al., 2021), (RHANOUI et al., 2019), (ZHOU; SUN; LIU; LAU, 2015) em problemas de análise de sentimentos existentes na literatura;
- A eficácia averiguada deste tipo de rede neural no processamento dados sequenciais, como textos, e em problemas específicos de classificação de textos, conforme afirma Jurafsky e Martin (2014).

A partir da criação de modelos de aprendizado de máquina construídos neste artigo, as empresas podem incorporá-los em seus sistemas de indicadores. Com isso, estes modelos podem auxiliar no acompanhamento de sua reputação e na tomada de decisão em relação às suas políticas de recrutamento e seleção de profissionais.

D. Organização do Artigo

Este artigo está organizado em 5 seções. Na seção II apresenta-se a fundamentação teórica dos conceitos e técnicas utilizadas em todo o artigo. Partindo desde concepções relacionadas à área de marketing, que fundamentam os conceitos de marca e auxiliam na definição dos objetivos a serem atingidos, até a conceituação técnica da área de Inteligência Artificial e suas metodologias para geração dos modelos de aprendizado de máquina.

Sequencialmente, na seção III são descritos os materiais e métodos aplicados no contexto deste artigo, partindo desde a etapa de extração e análise exploratória dos dados até as arquiteturas propostas e o treinamento dos modelos. Nas seções IV e V, respectivamente, são analisados e apresentados os resultados obtidos com o estudo, assim como, são exploradas as limitações, resultados e sugestões de melhorias futuras deste trabalho.

II. Fundamentação Teórica

Nesta seção são apresentados os conceitos teóricos que fundamentam as técnicas aplicadas no artigo.

A. Employee Value Proposition (EVP)

Desde a década de 90, a relação empregado e empregador assumiu uma visão na qual os empregados são clientes internos de uma organização. Neste sentido, mercados de alta competitividade por profissionais, necessitam realizar investimentos estratégicos para atrair e reter mão-de-obra qualificada (KALINSKA KULA; STANIEC, 2021).

Segundo Armstrong e Taylor (2014), uma prática que pode ser adotada pelas organizações para alcançar estes objetivos é a oferta de uma proposta de valor para o colaborador, também conhecido como *Employee Value Proposition* (EVP). É com base na EVP que a organização oferta aquilo que profissionais em prospecção e colaboradores existentes valorizam, desde fatores financeiros até outros fatores, também críticos, não ligados a este aspecto, os quais serão capazes de convencê-los a se juntar ou manter-se com a empresa. Com isso, espera-se que a organização seja reconhecida como uma instituição em que as pessoas queiram trabalhar e permanecer (ARMSTRONG; TAYLOR, 2014).

B. Employer Branding (EB)

Baseada na EVP, a organização pode apresentar sua oferta para seu público-alvo por meio de uma marca empregadora, também denominada como *Employer Branding* - EB (ARMSTRONG; TAYLOR, 2014). Segundo a Associação Americana de Marketing, uma marca ou *brand* é "um nome, termo, design, símbolo, ou qualquer outra característica que identifique bens ou serviços de um vendedor como distintos de outros vendedores". Desta forma, uma marca permite aos consumidores identificarem uma fonte ou alguém que produz um produto, que é algo que satisfaz uma necessidade ou desejo de um mercado. Além disso, a marca também funciona como um atalho para o consumidor organizar seu conhecimento e relacionar sua satisfação com o produto ou serviço e o auxiliar na sua tomada de decisão (KELLER, 2013).

Marcas também são utilizadas para identificar empresas. Quando estes princípios de marca são aplicados na criação de uma marca corporativa, que envolve o gerenciamento de recursos humanos, isto é chamado de *employer branding* (KELLER, 2013). Backhaus e Tikoo (2004) definem este termo como um processo de construção de uma identidade empregadora única, identificável e o que permite a sua dissociação dos seus concorrentes. Já Amber e Barrow (1996, pg. 187) a definem como "um pacote de benefícios funcionais, econômicos e psicológicos fornecidos pelo emprego, e identificados com a marca empregadora". Sua função consiste em sintetizar as relações da organização com seus colaboradores e com a identidade que ela apresenta ao mundo exterior (AMBER; BARROW, 1996).

Partindo para uma visão prática, a marca empregadora busca promover para o público interno e externo da organização uma visão clara dos seus diferenciais e os fatores que a tornam desejável. A partir dela, as pessoas criam imagens e desenvolvem associações da marca baseada nas informações disponíveis sobre a organização, as quais possuem impactos diretos sobre a atratividade, lealdade e produtividade dos profissionais, como representado no $framework$ da Figura 1, descrito por Backhaus e Tikoo (2004).

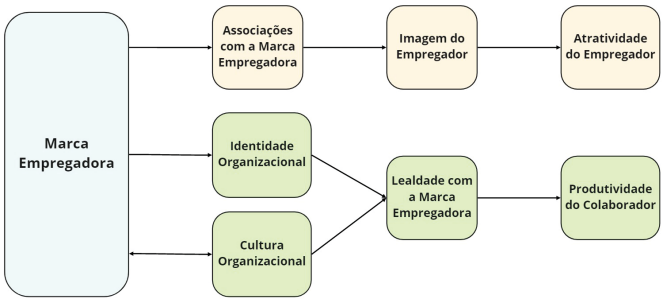


Figura 1. *Employer Branding framework*. Fonte: Elaborado pelo autor.

Como demonstrado nesta figura, a marca empregadora gera dois recursos: a associação e lealdade da marca. O primeiro molda a imagem empregadora e afeta diretamente a atratividade da organização para novos colaboradores, e permite melhorar seus índices de contratações. Enquanto o segundo, possui relação direta com sua identidade e cultura, os quais afetam diretamente o senso de pertencimento do colaborador com a organização e, consequentemente, a sua lealdade com a marca e permanência na empresa (BACKHAUS; TIKOO, 2004).

Para promover esta permanência ou retenção de profissionais, Armstrong e Taylor (2014) afirmam que as empresas precisam estabelecer estratégias de retenção. Para isso, as organizações devem partir da medição da taxa de pessoas que deixam a organização (taxa de $turnover$) e os custos incorridos do $turnover$ a fim de prever futuras perdas. A partir desta base, então, identificar os riscos que motivam os colaboradores a deixarem a organização e, assim, estabelecer formas de combatê-los (ARMSTRONG; TAYLOR, 2014).

Entretanto, as informações essenciais para construção e manutenção do EB nem sempre estão sob o poder das próprias organizações (Backhous and Tikoo, 2004). Neste sentido, é importante que as empresas estejam atentas às informações publicadas online a seu respeito, como em sites de recrutamento e seleção de profissionais semelhantes ao Glassdoor (SHANDWICK; RESEARCH, 2017). Pesquisas demonstram que 84% dos profissionais em busca de um emprego mudariam para uma empresa com uma boa reputação, enquanto 69% afirmaram que, mesmo desempregados, não estariam propensos a aceitar uma oferta de uma empresa com má reputação (Glassdoor, 2014). Por isso, para criar e manter a reputação da EB, as organizações devem ouvir atentamente seus colaboradores e priorizar ações consistentes para a construção da reputação da sua marca empregadora (GPTW, 2020).

C. Mineração de Textos

Segundo Gonçalves (2012), desde os anos 90 já existia uma clara percepção de que a grande parte da informação mundial encontrava-se em textos digitais, como por exemplo: páginas web, e-mails, documentos PDF, blogs, etc. Somente no período de 2001 a 2009 o número de páginas web cresceu 40% ao

ano, saltando de 10 milhões para 150 bilhões, sem incluir páginas privadas de corporações (MINER et al., 2012). Este fenômeno também é recente, já que o crescimento de textos digitais acompanha a evolução das redes sociais, tecnologias de transcrição de áudios, a disponibilização de conteúdos e digitalização de registros legados pelas organizações e etc (KWARTLER, 2017). Algumas motivações disto devem-se ao fato de que o formato textual é um dos meios mais intuitivos de se registrar conhecimentos, ideias, sentimentos e opiniões. Assim como, é normalmente gerado, utilizado e armazenado digitalmente em sistemas de informação (ARANHA, 2007). Neste sentido, seu crescimento em quantidade e complexidade acompanha a evolução da capacidade de processamento e armazenamento computacional (KWARTLER, 2017).

Esta imensidão de dados despertou, inicialmente, o interesse por parte de pesquisadores e empresas em analisar e descobrir novas informações a partir de textos digitais para utilizar em estratégias e tomadas de decisão (GONÇALVES, 2012). Entretanto, ao passo que o volume de dados torna-se promissor, este também gera dificuldades de gerenciamento de informações, principalmente, as de caráter não estruturado (ARANHA, 2007). Estima-se que dados textuais, que encontram-se em formatos semi-estruturados ou não-estruturados, representam, 75% e 80% dos dados no planeta, respectivamente. Neste sentido, o primeiro tipo de formato é caracterizado por possuir alguma estrutura definida, como documentos JSON ou XML (GONÇALVES, 2012). Já o segundo, não possui estrutura, como: livros, e-mails, páginas web, etc (MINER et al., 2012). Desta maneira, a coleta, transformação, análise e sumarização de ambos os formatos é mais complexa do que se comparada a dados textuais estruturados, como os encontrados em bancos de dados relacionais, por exemplo (GONÇALVES, 2012). Além disso, vale ressaltar que o tempo de análise de informações textuais aumenta proporcionalmente em relação ao crescimento do seu volume (ARANHA, 2007). Neste sentido, dada a dificuldade de manipular estes tipos de dados e a sua abundância, se faz necessário a utilização de processos automatizados para auxiliar os humanos na sua compreensão, exploração (MINER et al., 2012) e no ganho de eficiência na descoberta de conhecimento a partir dos textos (MORAIS, 2007).

Para suprir esta necessidade, surge a mineração de texto, ou *text mining*, que é uma subárea da área de "Descoberta de Conhecimento Apoiada por Computador", ou *Knowledge Discovery* (KD), que busca prover novos conhecimentos a partir de grandes volumes de dados (MORAIS, 2007). A área de mineração de texto também compartilha desse objetivo, porém com foco na análise e processamento de dados textuais (MINER et al., 2012). Conforme afirma Miner et al. (2012) esta área possui diversas intersecções com outros campos de conhecimento, que podem ser observadas na Figura 2, e que foram responsáveis por originá-la (ARANHA, 2007). Seu objetivo consiste, portanto, em utilizar mecanismos automáticos para descobrir conhecimento valioso registrado em textos ou conjunto de documentos (GONÇALVES, 2012), e assim fornecer vantagens para seus interessados (ARANHA, 2007).

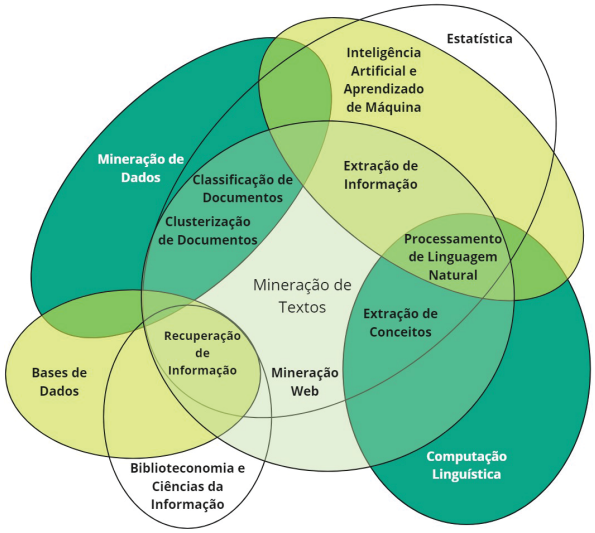


Figura 2. Diagramas de Venn que demonstram a relação da área de Mineração de Textos e os campos de conhecimento que a interseccionam. Fonte: Elaborado pelo autor.

Na prática, estas técnicas e processos podem ser usados em qualquer cenário em que texto seja uma entrada de dados. Por exemplo, para compreensão analítica de transcrições de textos acadêmicos (i), inteligência competitiva com base em revisão externa de textos (ii), revisão de pesquisas de satisfação com clientes (iii), tomadas de decisão baseada em dados (iv) e entre outras aplicações (KWARTLER, 2017).

Entretanto, para atingir os resultados almejados, como os apresentados nas aplicações práticas, anteriormente, é necessário percorrer um processo de mineração de textos. Afinal, os dados utilizados possuem um caráter, em geral, não estruturado e dinâmico, o que contrasta diretamente com os estruturados e armazenados em bancos de dados (ARANHA, 2007). Neste sentido, o processo de mineração de texto é um meio para levar vastas quantidades de dados textuais a um patamar estruturado que permita a extração de *insights* (KWARTLER, 2017). Desta forma, envolvendo todas as etapas necessárias para, desde o início, coletar, pré-processar, minerar e, finalmente, disponibilizar os dados para a análise do usuário (ARANHA, 2007).

D. Web Scraping

Para coletar dados a partir de páginas web, conforme aponta Miner et al. (2012), podem ser utilizadas técnicas e métodos da área de *web mining*. Este campo de conhecimento é responsável por minerar textos provenientes de páginas web. Para realizar este procedimento utiliza-se um *web crawler*, que é um tipo de sistema desta área que busca conteúdos na Internet. Como estes dados residem nas páginas web, os *web crawlers* podem utilizar um método, dentre os diversos existentes, conhecido como *web scraper*. Este método utiliza uma série de regras simples programadas por usuários para guiar o processo de busca e extração do conteúdo desejado na página web alvo (MINER et al., 2020).

Embora fundamental, a tarefa de coleta de dados é essencialmente trabalhosa e desafiadora (ARANHA, 2007). Na prática, os *web scrapers* realizam o trabalho dispendioso de extrair os dados das páginas web e constituir a base de dados para o restante do processo de mineração de textos. Desta forma, necessita-se somente da intervenção humana para programá-los a partir do local das páginas web e os dados de interesse a serem extraídos.

E. Inteligência Artificial e Aprendizado de Máquina

A Inteligência Artificial (IA), dentre as inúmeras definições existentes deste campo de conhecimento, é uma área que constrói e estuda como funcionam entidades inteligentes (RUSSELL; NORVIG, 2010). Para um sistema ser inteligente, ele deve saber aprender para se adaptar a ambientes de constante mudança. Caso contrário, seria necessário prever todas as situações possíveis (dados de entrada) e programar soluções para que o sistema conseguisse agir diante delas (dados de saída). Entretanto, na maioria dos casos não existe um passo a passo ou algoritmo, que explique como se comportar diante de todas as adversidades do ambiente. Mesmo que seja possível descobrir padrões ou certas regularidades que venham auxiliar na construção de uma aproximação que instrua o comportamento ideal (ALPAYDIN, 2014). Na prática, esta aproximação pode ser encontrada em um dos tipos de aprendizado de máquina existentes, que possui uma vasta aplicabilidade, sendo este o aprendizado indutivo.

Este tipo de aprendizado consiste em assimilar uma função ou regra que, baseada num conjunto de pares de dados de entrada e saída, consegue predizer saídas de forma autônoma, a partir de novos dados de entrada (RUSSELL; NORVIG, 2010). Como resultado disto, se obtém um modelo formado por parâmetros numéricos, os quais possibilitam gerar inferências a partir dos dados. Portanto, o aprendizado de máquina consiste em programar computadores para otimizar o desempenho destes modelos utilizando dados de experiências passadas (ALPAYDIN, 2014) a fim de se resolver problemas específicos (CASTRO; FERRARI, 2016)

Neste sentido, a escolha de quais técnicas e algoritmos são ideais para atender um problema depende de qual tipo de tarefa pretende ser realizada para atingir os objetivos almejados. Dentre as principais tarefas de mineração de dados existentes (GOLDSCHMIDT; PASSOS, 2005), destaca-se para a compreensão deste artigo a tarefa de classificação.

F. Classificação

Baseada em predição, a tarefa de classificação consiste em atribuir uma classe (valor discreto) a um objeto não conhecido que não possua rótulo ou classe pré-definida (CASTRO; FERRARI, 2016). A habilidade de atribuição provém da utilização de um modelo de conhecimento treinado, ou seja, que aprendeu antecipadamente com objetos já rotulados (MINER et. al, 2012). Este momento de preparação do modelo é conhecido como a etapa de treinamento, a qual é responsável por criar o classificador. Após ser gerado, o classificador passa pela etapa de testes, cujo objetivo consiste em avaliar o desempenho do modelo a partir de dados rotulados, que são diferentes dos utilizados na etapa de treinamento (CASTRO; FERRARI, 2016).

Neste sentido, compreende-se que, inerentemente, os classificadores adotam o tipo de aprendizado supervisionado, pois, é o ambiente (ou os dados) que ensinam o algoritmo de aprendizado. Este algoritmo encontra uma função capaz de mapear dados de entrada e de saída, utilizando como base exemplos já conhecidos em que, dado um conjunto de treino de *N* exemplos de pares de entrada-saída $(x_1, y_1), (x_2, y_2), ...(x_N, y_N)$, onde cada y_j foi gerado por uma funcão desconhecida *y = f(x)*, descubra uma função h que se aproxime da real função f (RUSSELL; NORVIG, 2010).

Nesta descrição é possível compreender que o aprendizado ocorre por meio da procura da melhor função ou hipótese *h*, dentro de um espaço de possíveis hipóteses, que mapeie corretamente um valor y diante de novos exemplos de dados *x* (RUSSELL; NORVIG, 2010). Neste sentido, existem algumas formas de avaliar o desempenho de um classificador. Uma delas é por meio da acurácia ou a precisão da hipótese *h* em mapear um dado de entrada *x* em *f(x)* (GOLDSCHMIDT; PASSOS, 2005). Em outras palavras, a acurácia é uma métrica que reflete o percentual ou proporção da classificação correta de registros. De forma complementar a acurácia, também é avaliada a taxa de erro do classificador, a qual consiste num indicador que mede inversamente a acurácia do modelo (GOLDSCHMIDT; PASSOS, 2005). De maneira geral, uma boa hipótese *h* é capaz de generalizar, ou seja, de predizer valores *y* a partir de novos exemplos (RUSSELL; NORVIG, 2010).

Entretanto, os conjuntos de treinamento são finitos e podem não ser suficientemente representativos (GOLDSCHMIDT; PASSOS, 2005). Além disso, estes conjuntos podem conter ruídos, como rótulos atribuídos erroneamente ou atributos escondidos que não foram levados em consideração no treinamento, por exemplo (ALPAYDIN, 2016). Em tais cenários o classificador pode ajustar excessivamente os parâmetros do modelo durante o treinamento, gerando *overfitting* (GOLDSCHMIDT; PASSOS, 2005). Este comportamento ocorre quando o modelo é complexo demais em relação ao tamanho e quantidade de ruído do conjunto de treinamento (GÉRON, 2017). Neste caso, o modelo apresenta uma baixa capacidade de generalização (CASTRO; FERRARI, 2016).

Outro fenômeno comum, que ocorre quando o modelo ajusta-se insuficientemente aos dados de treinamento, é chamado de *underfitting* (GOLDSCHMIDT; PASSOS, 2005). Isto ocorre quando o modelo é simples demais para aprender com a complexidade inerente dos dados existentes (GÉRON, 2017). Neste caso, o modelo não é capaz de representar o conjunto de dados. Portanto, um bom modelo é aquele que consegue obter equilíbrio entre estes dois extremos, ou seja, que apresenta flexibilidade durante o treinamento, ao passo que não interpola os ruídos do conjunto de dados (CASTRO; FERRARI, 2016). Na Figura 3, conforme demonstra Castro e Ferrari (2016), é possível observar em sequência a representação cartesiana de três modelos. Sendo o exemplo (a) uma representação de

um modelo com *underfitting*, o (b) representando um modelo com *overfitting* e o (c) um modelo ideal. Nestas exemplos, os pontos representam os pares de entrada e saída de dados no plano cartesiano, e a linha o modelo treinado.

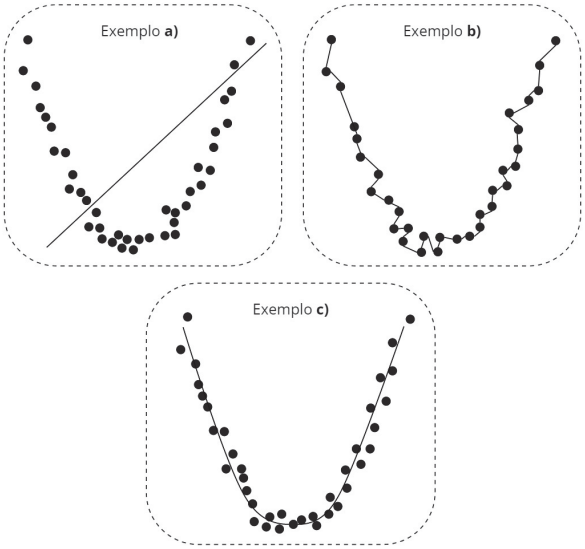


Figura 3. Fonte: Elaborado pelo autor.

Para auxiliar no alcance de um modelo equilibrado, pode-se aplicar a validação cruzada ou *cross-validation*, que consiste num método sistemático capaz de encontrar o momento ideal de parada do treinamento do classificador (CASTRO; FERRARI, 2016). Neste método, o conjunto de treinamento é dividido em duas partes, uma focada no treinamento e outra na validação. O objetivo do conjunto de validação é testar a generalização do modelo (ALPAYDIN, 2016). Desta forma, a validação cruzada avalia recorrentemente a taxa de erro do modelo de treinamento diante do conjunto de validação. O ponto de parada ocorre quando esta taxa começa a aumentar consecutivamente no conjunto de validação, indicando uma possível tendência de deterioração da generalização, desta forma, obtendo-se os parâmetros do modelo que melhor generalize (CASTRO; FERRARI, 2016).

Outra maneira de auxiliar na busca de um modelo com boa performance, é a utilização da técnica de *grid search*. Esta técnica, que pode ser combinada com o *cross validation*, automatiza a busca da melhor combinação de hiperparâmetros que melhor ajustem os modelos. Desta forma, não só o trabalho manual é eliminado, como a possibilidade de testar uma ampla gama de combinações é possibilitada, de forma organizada (GÉRON, 2017).

De forma geral, modelos de classificação possuem uma vasta aplicabilidade. No contexto da mineração de textos são comumente aplicados na classificação de *spam*, documentos judiciais, análise de sentimento em canais digitais e entre outras aplicações (MINER et. al, 2012). Destacando a última, a análise de sentimento tem sido amplamente utilizada por organizações para aprofundar o conhecimento sobre seus clientes (MARTINS et al., 2020).

G. Análise de Sentimento

A capacidade de saber sobre as percepções das pessoas a respeito de algo é um recurso informacional importante para auxiliar em processos de tomada de decisão (MINER et al., 2012). Afinal, conhecer as opiniões de outras pessoas é um aliado neste processo, tanto para empresas, quanto para os indivíduos em geral (INDURKHYA; DUMERAU, 2010). Por exemplo, antes de fazer uma grande compra, uma pessoa normalmente consulta avaliações de outros consumidores na Internet para tomar sua decisão (MINER et al., 2012). Assim como, empresas também buscam identificar a opinião do seu público alvo para entender a qualidade, a aceitação de seus produtos ou serviços ou até mesmo investigar a relevância de sua concorrência no mercado. No passado, para ambos os lados, este processo de descoberta de opinião era diferente. Os consumidores, normalmente, consultavam a opinião de pessoas a partir de seus círculos de relacionamento ou de especialistas (MARTINS et al., 2020). Enquanto as empresas dependiam de processos formais de pesquisas de opinião, enquetes e grupos de foco.

Em geral, este cenário passou a mudar após o crescimento exponencial da Web e das publicações de conteúdo geradas por usuários em blogs, sites, fóruns de discussão e etc. A Web facilitou este movimento e, inclusive, mudou a forma como as pessoas expõem suas opiniões (INDURKHYA; DUMERAU, 2010). Neste sentido, as pessoas passaram a publicar constantemente seus comentários, sentimentos, experiências a fim de expressar sua satisfação ou insatisfação por um objeto, serviço, evento ou algo que lhe interesse (MARTINS et al., 2020).

No entanto, a facilidade que a web trouxe também foi responsável por gerar novos desafios. Diante do imenso volume de textos expressando opiniões e sentimentos, assim como a pluralidade de fontes que detém essas informações, tornou-se difícil para os seres humanos lidarem com todo este contexto. Afinal, as dificuldades variam desde encontrar opiniões em textos imersos e espalhados em fontes variadas, até ler, resumir e organizar cada texto no formato adequado para análises posteriores. Diante deste contexto, surge a necessidade de utilizar sistemas para auxiliar neste processo, os quais surgem da área de análise de sentimentos, também conhecida como mineração de opiniões (INDURKHYA; DUMERAU, 2010). Esta área estuda técnicas computacionais para localizar e extrair informações subjetivas em textos, como por exemplo, opiniões (MARTINS et al., 2020). Em sua essência, opinião é uma expressão subjetiva de um indivíduo refletida em forma de sentimentos, apreciações e sensações em direção a uma entidade e suas características (INDURKHYA; DUMERAU, 2010). Desta forma, conforme cita Indurkhya e Dumerau (2010, pg. 633) “a análise de sentimento busca inferir os sentimentos das pessoas com base nas suas expressões linguísticas”.

Para alcançar os resultados propostos pela análise de sentimentos, a área dispõe de diferentes abordagens que podem ser aplicadas de acordo com o problema existente, conforme apresenta Indurkhya e Dumerau (2010, pg. 628-629). Uma

dessas abordagens e que, também, possui sinergia com o presente artigo, é a tarefa de classificação de sentimento, a qual trata a análise de sentimento como um problema de classificação (INDURKHYA; DUMERAU, 2010).

Dado que "a análise de sentimento é o processo de extração da intenção emocional do autor de um texto" (KWARTLER, 2017, pg. 85), a classificação de sentimentos busca encontrar no texto o sentimento do autor e classificá-lo de forma binária (ex: negativo ou positivo), ou com mais de duas classes (ex: negativo, neutro, positivo) (JURAFSKY; MARTIN, 2014). De forma geral, a maioria das técnicas de classificação de sentimento utiliza o aprendizado supervisionado para concretizar o seu objetivo (INDURKHYA; DUMERAU, 2010).

H. Redes Neurais

Segundo Russel e Norvig (2010), redes neurais (ou *Neural Networks* - NNs) são essencialmente grupos de unidades ou neurônios conectados, que possuem propriedades determinadas por sua topologia e as características dessas unidades. Seu objetivo consiste em processar valores observados a fim de gerar predições (MINER et al., 2012). Neste sentido, cada neurônio atua como uma unidade de processamento que recebe um conjunto de valores de entrada, os processa e gera um resultado (JURAFSKY; MARTIN, 2014). Tanto para receber quanto para propagar estes valores, os neurônios dependem, respectivamente, de conectores de entrada e saída, conforme o modelo matemático neural representado na Figura 4 (RUSSELL; NORVIG, 2010).

No centro desta figura, é possível observar um neurônio *j* (representado por uma elipse) que, dentre os diversos conectores de entrada (representados por flechas), propaga um sinal ou peso numérico w_{ij} de um neurônio *i* para o *j* que corresponde a sua força de conexão. Vale ressaltar, que um neurônio pode receber valores *w* de *n* neurônios dentro de uma rede (RUSSELL; NORVIG, 2010).

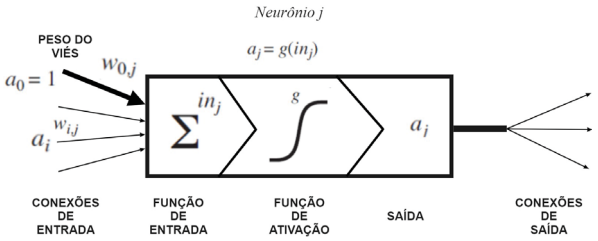


Figura 4. A representação de um neurônio j em um modelo. Fonte: Elaborado pelo autor.

Ao receber os valores de entrada, o primeiro passo do processamento consiste em aplicar uma função de entrada in_j para realizar a soma ponderada dos valores *w*, adicionando o viés $a_0 = 1$ associado a um peso $w_{0,j}$ conforme é representado na Equação 1 (RUSSELL; NORVIG, 2010):

$$in_j = \sum_{i=0}^{n} w_{i,j} a_i \tag{1}$$

Sequencialmente, aplica-se uma função não linear de ativação g para garantir a propriedade de não linearidade dos neurônios da rede (RUSSELL; NORVIG, 2010), conforme a Equação 2:

$$a_j = g\left(in_j\right) = g\left(\sum_{i=0}^{n} w_{i,j} a_i\right) \tag{2}$$

Neste ponto, cabe enfatizar a existência de diversos tipos de funções de ativação. Dentre os tipos mais populares, destacam-se as funções Sigmóide, Softmax e ReLU (JURAFSKY; MARTIN, 2014).

A função de ativação sigmóide é uma função que mapeia valores reais em um intervalo [0, 1] (JURAFSKY; MARTIN, 2014), descrita pela Equação 3:

$$y = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + \exp(-z)} \tag{3}$$

Por causa da sua característica achatada perto em suas extremidades e linear próximo a zero, como representada na Figura 5, esta função tende a "espremer" valores outliers nas extremidades do seu intervalo [0,1].

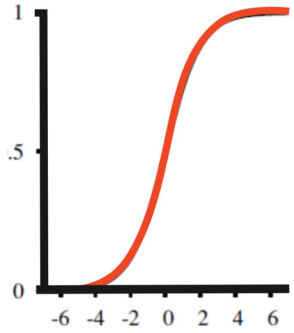


Figura 5. Função de Ativação Sigmóide. Fonte: Elaborado pelo autor.

Outra função de ativação comum é a Softmax. Trata-se de função exponencial que, dada uma instância *x*, computa a probabilidade *p* de um valor *y* pertencer a uma determinada classe *c* em problemas não-binários, conforme a seguinte a representação (JURAFSKY; MARTIN, 2014):

$$p(y = c|x)$$

Esta função mapeia uma distribuição de probabilidade para um vetor $z = [z_1; z_2; ...; z_k]$ de *k* valores arbitrários em um intervalo entre 0 e 1, e é descrita pela Equação 4:

$$\text{softmax}\left(z_i\right) = \frac{\exp\left(z_i\right)}{\sum_{j=1}^{k} \exp\left(z_j\right)} 1 \leq i \leq k \tag{4}$$

Entretanto, assim como a função Sigmóide, a função Softmax também é impactada pelos problemas de dissipação e explosão de gradiente, pois valores elevados de in_j resultam em valores de saída saturados (JURAFSKY; MARTIN, 2014), ou seja, extremamente próximos a 1 (explosão de gradiente) ou com derivadas muito próximas a 0 (dissipação de gradiente)

(GÉRON, 2017). Além disso, conforme mencionam Russel e Norvig (2010) e Jurafksy e Martin (2014), as funções Sigmóide e Softmax são geralmente utilizadas na última camada da rede neural.

Por fim, a função de ativação ReLU (*Rectified Linear Unit*) é uma função quase linear que preserva as propriedades inerentes da otimização de modelos lineares (GOODFELLOW; BENGIO; COURVILLE, 2016). Esta função mantém valores x, quando x é positivo, e altera para 0 quando são negativos, de acordo com a Equação 5. Além disso, seu comportamento pode ser observado no plano cartesiano da Figura 6 (JURAFSKY; MARTIN, 2014).

$$f(x) = \max(0, x) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{cases} \tag{5}$$

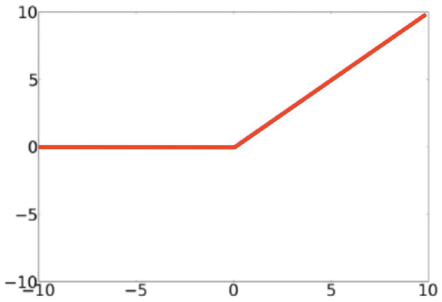


Figura 6. Função de Ativação ReLU. Fonte: Elaborado pelo autor.

Deste modo, a função ReLU não possui problemas com dissipação de gradiente, pois a sua derivada nos casos de valores de pesos elevados é 1 (JURAFSKY; MARTIN, 2014). Outra característica relevante desta função, é sua utilização comum nas camadas internas das redes neurais, conforme afirmam Russel e Norvig (2010) e Jurafksy e Martin (2014).

Depois de aplicada a função de ativação, obtém-se a saída, ou valor ativado, do neurônio a_j (JURAFSKY; MARTIN, 2014). Este valor pode ser compreendido pela Equação 6.

$$a_j = g\left(\sum_{i=0}^{n} w_{i,j} a_i\right) \tag{6}$$

Nesta equação o valor a_i é a saída do neurônio i e $w_{i,j}$ o peso do conector com origem de i para j (RUSSELL; NORVIG, 2010). Desta forma, a partir destas conceituações, é possível compreender o funcionamento individual de um neurônio. A atuação em conjunto destas unidades formam as redes neurais (JURAFSKY; MARTIN, 2014).

Neste sentido, na Figura 4 pode-se observar que os conectores de entrada e de saída do neurônio j apontam para uma única direção. Deste modo, a criação de uma rede neural em que as conexões se comportam desta forma, como um grafo acíclico direto, são denominadas como redes *feed-forward* (RUSSELL; NORVIG, 2010).

Este tipo de rede neural não armazena nenhum tipo de estado em si ou atua de forma cíclica como as redes neurais recorrentes apresentadas na seção II-I (JURAFSKY; MARTIN, 2014). Além disso, sua estrutura é composta por neurônios organizados em camadas (*layers*), como pode ser observado na Figura 7, em cada neurônio recebe entradas a partir da camada anterior de forma contínua (RUSSELL; NORVIG, 2010).

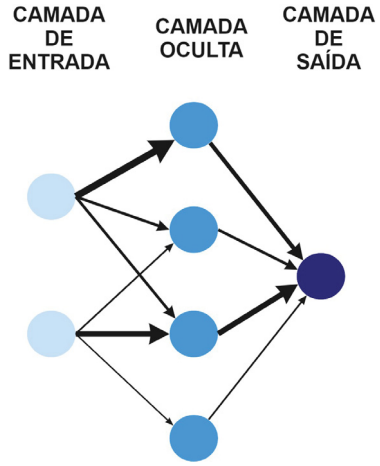


Figura 7. Rede Neural Simples. Fonte: Elaborado pelo autor.

Nesta imagem, além das camadas de entrada (*input layer*) e saída (*output layer*), é possível observar a existência de uma camada oculta (*hidden layer*). Este tipo de camada contém neurônios chamados "ocultos" e que se comportam como o apresentado na Figura 4, na qual cada unidade realiza uma soma ponderada dos valores de entrada e aplica uma função de ativação para gerar uma saída (JURAFSKY; MARTIN, 2014).

No caso da Figura 7, a camada oculta também é totalmente conectada (*full-connected*), ou seja, todos os neurônios da camada oculta estão conectados com todos os neurônios da camada anterior. Além disso, esta rede possui apenas uma camada oculta, cujas saídas dos neurônios são propagadas diretamente para a camada de saída. Entretanto, vale ressaltar, que existem redes que possuem mais de uma camada oculta em sua estrutura, como as redes multicamadas (*multi-layer networks*) (JURAFSKY; MARTIN, 2014).

Nestas redes, as saídas de neurônios das camadas ocultas não se propagam diretamente para a camada de saída da rede neural. Desta maneira, a existência de inúmeras camadas ocultas remete à "profundidade" das redes neurais modernas, as quais são reconhecidas por esta característica e sua utilização denominada como aprendizado profundo (*deep learning*) (JURAFSKY; MARTIN, 2014).

De forma ampla, uma rede neural *feed-forward* pode ser visualizada como um exemplo de aprendizado supervisionado, no qual para cada observação x existe uma saída y correta. Na equação 7 é possível observar este comportamento através de uma rede neural com duas camadas, sendo o número de camada representado entre colchetes (JURAFSKY; MARTIN, 2014).

Nesta equação, x é representado por $a^{[0]}$, o qual se refere à camada de entrada da rede neural, sendo $a^{[i]}$ a saída de uma camada i; $z^{[i]}$ como a combinação de pesos $W^{[i]}a^{[i-1]} + b^{[i]}$;

e g uma função de ativação. Por fim, $\hat{y}$ representa a estimativa da rede quanto ao y correto esperado. Desta forma, o objetivo de todo este procedimento é treinar a rede neural para aprender os parâmetros $W^{[i]}$ e o viés $b^{[i]}$ para cada camada i a fim de obter o resultado $\hat{y}$ mais aproximado do resultado verdadeiro y (JURAFSKY; MARTIN, 2014).

$$\begin{aligned} z^{[1]} &= W^{[1]}a^{[0]} + b^{[1]} \\ a^{[1]} &= g^{[1]}\left(z^{[1]}\right) \\ z^{[2]} &= W^{[2]}a^{[1]} + b^{[2]} \\ a^{[2]} &= g^{[2]}\left(z^{[2]}\right) \\ \hat{y} &= a^{[2]} \end{aligned} \tag{7}$$

Dado o objetivo de se descobrir uma saída y a partir de uma observação e, modelar esta distância, se utilizada uma função de perda (*loss function*). A função de perda $L(x, y, \hat{y})$ é definida como a quantidade de utilidade (a desejabilidade de um resultado ou estado) perdida por prever $h(x) = \hat{y}$ quando as respostas corretas correspondem a $f(x) = y$ (RUSSELL; NORVIG, 2010).

Segundo Jurafsky e Martin (2014), uma função de perda comumente utilizada é a *cross-entropy loss*. Dado que y é um vetor que contém C classes que correspondem a saída correta esperada, esta função pode ser definida como a Equação 8, para um problema não-binário. Sua forma simplificada pode ser compreendida como um log da probabilidade de um resultado correspondente a uma determinada classe y, conforme a Equação 9 (JURAFSKY; MARTIN, 2014).

$$L_{CE}(\hat{y}, y) = -\sum_{i=1}^{C} y_i \log \hat{y}_i \tag{8}$$

$$L_{CE}(\hat{y}, y) = -\log \hat{y}_i, \quad \text{(onde } i \text{ é a classe correta)} \tag{9}$$

Para identificar os parâmetros que minimizem a perda da função *loss*, aplica-se um algoritmo de otimização de descida de gradiente. No entanto, para utilizar este algoritmo, é necessário identificar o gradiente da função de perda, que consiste num vetor que possui a derivada parcial da função de perda para cada um dos parâmetros (JURAFSKY; MARTIN, 2014).

Dada uma função $y = f(x)$, a derivada $f'(x)$ serve para determinar como uma pequena mudança num valor de entrada da função corresponde a uma alteração na sua saída, conforme $f(x+\epsilon) \approx f(x)+\epsilon f'(x)$. Desta forma, no contexto da função de minimização da função de perda, a derivada contribui para mostrar como uma alteração num parâmetro x pode refletir em uma melhoria numa saída y (*loss*) (GOODFELLOW; BENGIO; COURVILLE, 2016).

Para encontrar o ponto de menor perda, estabelece-se um ponto inicial (w_0, w_1) no espaço de pesos - um espaço definido por todas as configurações de pesos possíveis. Sequencialmente, este ponto se move para outro ponto descendente na vizinhança até se encontrar o ponto mínimo de perda, conforme descreve o pseudoalgoritmo da Figura 8 (RUSSELL; NORVIG, 2010). Neste pseudo algoritmo, o parâmetro α refere-se à taxa de aprendizado, que diz respeito ao tamanho do passo ou a distância no espaço que o ponto se move (GOODFELLOW; BENGIO; COURVILLE, 2016).

$\mathbf{W} \leftarrow$ qualquer ponto no espaço de parâmetros
Repita até convergir **faça**
  Para cada w_i **em W faça**
    $w_i \leftarrow w_i - \alpha \dfrac{\partial}{\partial w_i} Loss(\mathbf{w})$

Figura 8. Pseudoalgoritmo de descida de gradiente. Fonte: Elaborado pelo autor.

O objetivo geral desta busca, portanto, consiste em encontrar o menor valor absoluto de $f(x)$, o mínimo global, evitando pontos críticos como mínimos, máximos ou cumes locais, os quais limitam os passos e evitam que o mínimo global seja encontrado, como representado na Figura 9 (GOODFELLOW; BENGIO; COURVILLE, 2016).

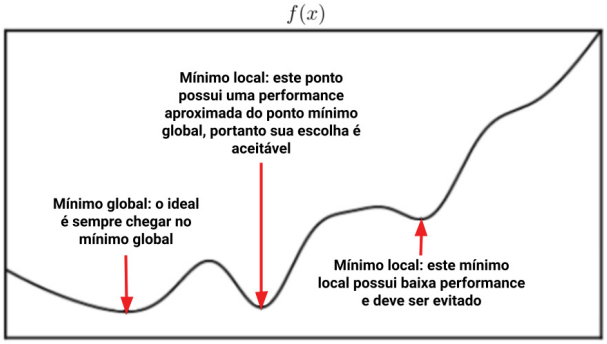


Figura 9. Ilustração dos conceitos envolvidos na descida de gradiente. Fonte: Elaborado pelo autor.

Dentre os diversos tipos de algoritmos de otimização existentes, Adam é um algoritmo para otimização de funções objetivas estocásticas, baseado em gradiente de primeira ordem. Este método utiliza como base uma estimativa do primeiro e segundo momento dos gradientes para computar taxas de aprendizado adaptativas para diferentes combinações de parâmetros (KINGMA; BA, 2015).

I. Redes Neurais Recorrentes

Conforme mencionado na seção II-H, as redes neurais do tipo *feed-forward* possuem conexões em apenas uma direção, formando grafos acíclicos diretos (RUSSELL; NORVIG, 2010). Diferentemente delas, redes neurais que possuem ciclos em suas conexões de rede são conhecidas como redes neurais recorrentes (RNR) ou *Recurrent Neural Networks* (RNN) (JURAFSKY; MARTIN, 2014). Estes ciclos representam o comportamento das RNNs ao utilizar suas próprias saídas como novas entradas (RUSSELL; NORVIG, 2010), que evidencia como um estado atual de um neurônio influencia seu próprio estado em um passo de tempo futuro. Neste sentido, as redes neurais recorrentes podem ser definidas como uma família de

redes neurais especializadas em processar dados sequenciais, como uma sequência de valores $x(1), ..., x(t)$, em que um índice t varia de 1 a um número finito t (GOODFELLOW; BENGIO; COURVILLE, 2016).

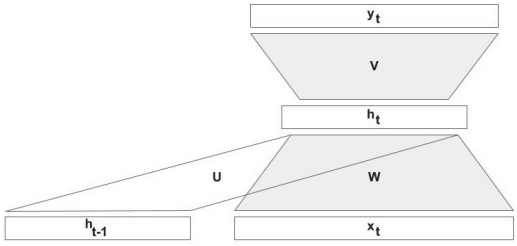


Figura 10. RNN simples. Fonte: Elaborado pelo autor.

Na Figura 10, é possível observar como uma inferência é realizada numa RNN simples. Nesta representação, inicialmente, a entrada de um vetor x_t é multiplicada por uma matriz de pesos W e aplicada uma função de ativação não-linear. Na sequência, ocorre o que justamente confere a recorrência para a rede, que consiste no procedimento de multiplicar a camada oculta h_{t-1} com uma matriz de pesos U. Desta forma, a camada oculta do passo anterior opera como um tipo de memória, que codifica o processamento anterior e comunica quais decisões devem ser tomadas em passos subsequentes. Tais valores calculados, então, são somados e passados por uma função de ativação g e propagados para uma camada a camada oculta h_t, que calculam o resultado de uma saída y_t (JURAFSKY; MARTIN, 2014). Tal processo introduzido é representado pela seguinte equação:

$$\begin{aligned} h_t &= g\left(Uh_{t-1} + Wx_t\right) \\ y_t &= f\left(Vh_t\right) \end{aligned} \tag{10}$$

Esta característica de computação recorrente pode ser compreendida como uma cadeia de eventos que desdobram em t espaços de tempo, compartilhando parâmetros ao longo da rede, como pode ser observado na Figura 11.

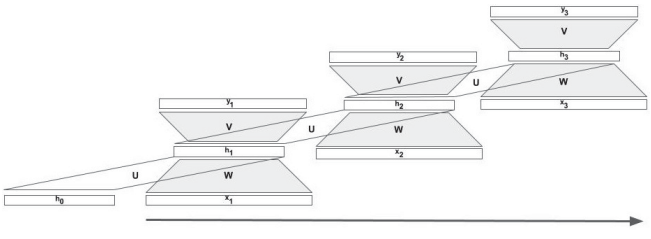


Figura 11. RNN com mais de um passo t. Fonte: Elaborado pelo autor.

Esta característica das RNNs confere a possibilidade de aprender um único modelo de conhecimento ao longo de toda a sequência de eventos t. Este comportamento pode ser compreendido por meio da Equação 11.

$$\begin{aligned} \boldsymbol{h}^{(t)} &= g^{(t)}\left(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(t-1)}, \boldsymbol{x}^{(t-2)}, \ldots, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(1)}\right) \\ &= f\left(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \boldsymbol{\theta}\right) \end{aligned} \tag{11}$$

Na Equação 11, um modelo de conhecimento representado pela função $g^{(t)}$ recebe uma sequência de valores passados $\left(\boldsymbol{x}^{(t)}, \boldsymbol{x}^{(t-1)}, \boldsymbol{x}^{(t-2)}, \ldots, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(1)}\right)$ para gerar um estado atual $h^{(t)}$. Neste sentido, ao aplicar repetidas vezes uma função f, consegue-se obter um único modelo de conhecimento f que compartilha os mesmo parâmetros ao longo de todos os passos possíveis t. Com isso, por exemplo, tais modelos podem ser treinados com menos instâncias e generalizar em ocasiões em que entradas de valores não estão presentes num conjunto de treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016).

Outra vantagem é que, pelo fato deste tipo de rede ser especializada em dados sequenciais, ela torna-se um tipo extremamente eficaz para o aprendizado de modelos envolvendo linguagem natural, como em textos, e sua aplicação em tarefas de classificação, como a análise de sentimento, por exemplo (JURAFSKY; MARTIN, 2014). Afinal, modelos baseados em RNNs visualizam textos como uma sequência de palavras e, desta forma, são capazes de capturar dependências entre palavras e estruturas textuais utilizadas em tarefas de classificação de textos (MINAEE et al., 2021).

J. Long Short-Term Memory (LSTM)

Dentre todas as vantagens das RNNs apontadas na seção II-I, sobretudo em sua forma convencional, cabe salientar que estas redes são marcadas por problemas decorrentes da sua própria estrutura.

Conforme afirma Jurafsky e Martin (2014), RNNs convencionais tendem a apresentar dificuldades em carregar informações de contextos que estão distantes do passo atual em processamento, que é uma capacidade crucial para aplicações envolvendo linguagem. Isto ocorre porque os contextos de informações codificadas nas camadas ocultas tendem a ser locais e mais úteis para decisões e processamento de parte de sequências mais recentes. Afinal, a informação contida nas camadas ocultas e suas conexões são úteis para tomadas de decisão atuais e futuras. Aliado a isso, existe o fato de que a perda de sinal de cada camada oculta em um passo t é propagado para as camadas seguintes. Somando estes fatores às multiplicações realizadas nas camadas ocultas, obtêm-se um erro comum das RNNs convencionais que é a dissipação de gradiente, que ocorre quando os gradientes encontram-se muito próximos de zero, gerando lentidão e problemas durante o treinamento das redes (JURAFSKY; MARTIN, 2014).

Neste sentido, algumas arquiteturas mais complexas baseadas em RNNs, como a LSTM *(Long Short-Term Memory)*, surgiram para resolver o problema de dissipação de gradiente sofrido pelas redes neurais recorrentes convencionais e capturar dependências de longa duração (MINAEE et al., 2021). Para isso, a LSTM divide o problema da perda de contexto entre: remover informações contextuais desnecessárias e adicionar informações relevantes para tomada de decisão futura, dando autonomia para a unidade realizar este gerenciamento (JURAFSKY; MARTIN, 2014). Esta habilidade provém da introdução de células LSTM ou de memória, que possuem um *loop* interno, aquém da recorrência da RNN (GOOD-

FELLOW; BENGIO; COURVILLE, 2016), para relembrar de informações em intervalos arbitrários, e três "portões" (*gates*) especializados para controlar o fluxo de saída e entrada de informações (MINAEE et al., 2021). Tais estruturas podem ser observadas na Figura 12.

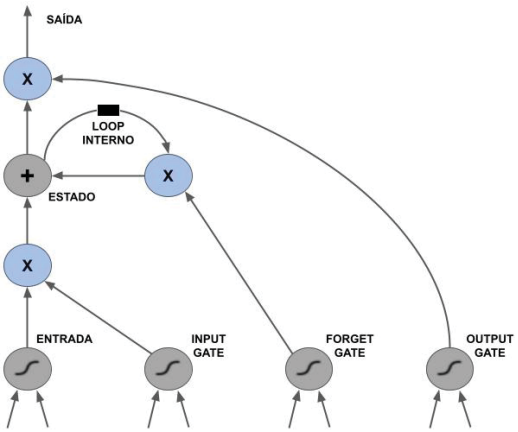


Figura 12. Estrutura de uma LSTM. Fonte: Elaborado pelo autor.

O *forget gate* é responsável por manter ou remover informações do contexto. Conforme a Equação 12 apresentada a seguir, este *gate* realiza uma soma ponderada dos estados das camadas ocultas anterior e atual, e a passa por uma função de ativação Sigmóide. O resultado desta função é multiplicado, então, pela informação do contexto que não é necessária. Além disso, também se computa a informação dos estados das unidades anterior e recente, conforme a Equação 13, que é padrão das RNNs (JURAFSKY; MARTIN, 2014).

$$\begin{aligned} f_t &= \sigma\left(U_f h_{t-1} + W_f x_t\right) \\ k_t &= c_{t-1} \odot f_t \end{aligned} \tag{12}$$

$$\begin{aligned} h_t &= g\left(U h_{t-1} + W x_t\right) \\ y_t &= f\left(V h_t\right) \end{aligned} \tag{13}$$

Já o *input gate* é responsável por selecionar as informações contidas no vetor do contexto atual, conforme a Equação 14, e adicionar o vetor de contexto modificado a um novo vetor, como descreve a Equação 15:

$$\begin{aligned} h_t &= g\left(U h_{t-1} + W x_t\right) \\ y_t &= f\left(V h_t\right) \end{aligned} \tag{14}$$

$$c_t = j_t + k_t \tag{15}$$

Por fim, o *output gate* atua como um decisor de qual informação é necessária para o estado oculto recente, apresentado na equação seguinte:

$$\begin{aligned} o_t &= \sigma\left(U_o h_{t-1} + W_o x_t\right) \\ h_t &= o_t \odot \tanh\left(c_t\right) \end{aligned} \tag{16}$$

K. Bi-directional Long Short-Term Memory (BiLSTM)

Rede neural recorrente bidirecional (BRNN) é um tipo de RNN proposta por Schuster e Paliwal (1997), em que duas RNNs independentes processam os dados de entrada em sentidos opostos. Desta forma, um dos sentidos processa as entradas do começo para o fim (*forward*), enquanto o outro atua do fim para o começo (*backward*), como é representado na Figura 13 (JURAFSKY; MARTIN, 2014).

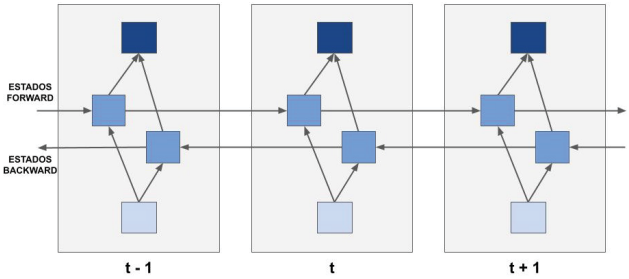


Figura 13. Estrutura de uma BiLSTM. Fonte: Elaborado pelo autor.

O seu objetivo consiste em adquirir mais conhecimento sobre o contexto ao capturá-lo em mais de uma perspectiva e, então, concatenar ambas as saídas em uma representação contextual única em cada tempo t. Este comportamento é demonstrado na notação a seguir, dado que h_t^f corresponde ao conhecimento de contexto obtido do começo até o passo t, e h_t^b o conhecimento obtido da direita até o passo atual (JURAFSKY; MARTIN, 2014).

$$\begin{aligned} h_t^f &= RNN_{\text{forward}}\left(x_1^t\right) \\ h_t^b &= RNN_{\text{backward}}\left(x_t^n\right) \\ h_t &= h_t^f \oplus h_t^b \end{aligned} \tag{17}$$

Neste sentido, uma rede *Bi-directional Long Short-Term Memory* (BiLSTM), ou simplesmente LSTM bi-direcional, consiste na combinação de duas LSTMs atuando com o comportamento de uma BRNN sobre os dados de entrada (SIAMI-NAMINI; TAVAKOLI; NAMIN, 2019). Com isso, espera-se que o melhor aproveitamento do contexto contribua para o aprimoramento da acurácia de modelos de conhecimento (GRAVES; FERNANDEZ; SCHMIDHUBER, 2005).

L. Redes Neurais Convolucionais

Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNNs) são redes neurais especializadas em processar dados bi-dimensionais, embora também possa ser utilizada para dados uni ou tridimensionais. Sua principal característica é a utilização de uma operação matemática linear chamada "convolução", a qual confere o nome da rede, em pelo menos uma de suas camadas (GOODFELLOW; BENGIO; COURVILLE, 2016).

O objetivo desta operação consiste em extrair características específicas de uma forma em que só a informação desejada seja recuperada, ao contrário das redes neurais completamente conectadas, desta forma, diminuindo o uso de memória e aprimorando a eficiência estatística (GOODFELLOW; BENGIO;

COURVILLE, 2016). Outra grande vantagem deste tipo de rede, é o fato de que uma vez que as CNNs aprendem a detectar um tipo de padrão, este padrão pode ser reconhecido em qualquer outro local (GÉRON, 2017), sendo este um grande aliado para predições.

Desta forma, a operação de convolução pode ser compreendida por meio de uma notação em que uma função h representa a convolução de duas funções f e g (RUSSELL; NORVIG, 2010), em que x é um índice de valores inteiros; f geralmente um *array* multidimensional ou um *tensor*, e g um $array$ bidimensional de pesos chamado de $kernel$ ou máscara. O resultado da convolução é denominado como um mapa de características ou *feature map* (GOODFELLOW; BENGIO; COURVILLE, 2016). A convolução representada por uma equação num contexto unidimensional é:

$$h(x) = (f * g)(x) = \sum_{u=-\infty}^{+\infty} f(u)g(x-u) \tag{18}$$

E num contexto bidimensional:

$$h(x,y) = (f * g)(x,y) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} f(u,v)g(x-u, y-v) \tag{19}$$

De forma prática, na Figura 14 é apresentado um exemplo em que a convolução é aplicada em um $array$ de duas dimensões. Nesta representação, um $kernel$ de tamanho 2x2 é aplicado em todas as posições do $array$ bidimensional de entrada ($input$), da esquerda para a direita e de cima para baixo, realizando um produto escalar entre os $arrays$. O resultado desta operação é o *feature map*, um novo $array$ bidimensional contendo as saídas ($output$) da convolução.

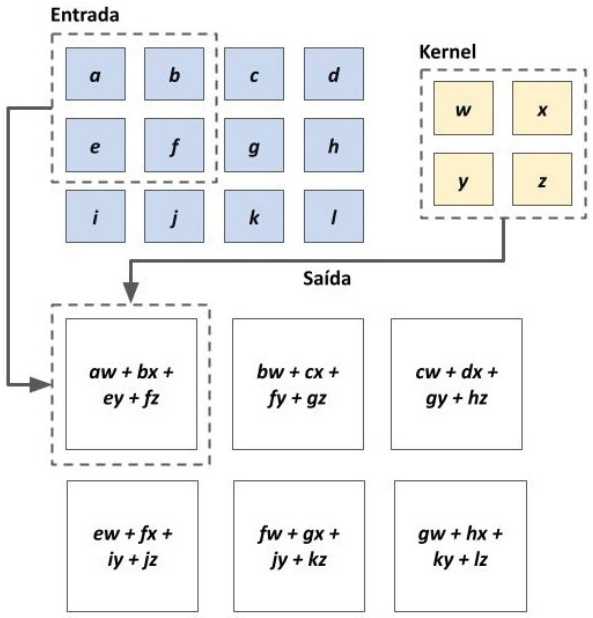


Figura 14. Comportamento de uma CNN. Fonte: Elaborado pelo autor.

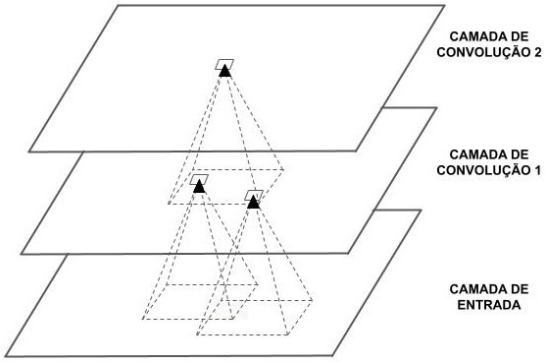


Figura 15. Camadas de convolução. Fonte: Elaborado pelo autor.

As convoluções nas CNNs estão presentes em camadas ocultas de convolução. Como na Figura 15, e simulando o comportamento descrito na Figura 14, os neurônios da primeira camada de convolução estão apenas conectados em alguns $pixels$ da camada de entrada. Assim como, os neurônios da segunda camada estão conectados em alguns neurônios da primeira. Com esta estrutura hierárquica, a rede consegue recuperar informações mais detalhadas a respeito das características desejadas à medida em que as camadas são acrescidas (GÉRON, 2017).

Além da convolução, outra camada exerce um papel fundamental nas CNNs: a de $pooling$ (Figura 16). A função desta camada consiste em tirar uma amostra encolhida de um dado de entrada para reduzir o uso de memória e custo computacional, assim como o número de parâmetros da rede a fim de evitar $overfitting$. Na prática, ele funciona de maneira similar ao comportamento das camadas de convolução representadas na Figura 15 (GÉRON, 2017).

Entretanto, a diferença é que os neurônios da camada de $pooling$ não possuem pesos atrelados, desta forma, apenas realizam funções de agregação como max ou média sobre uma determinada outra camada utilizando $kernels$ para isso, conforme representado na Figura 16. Nesta figura, é representada um tipo de camada de $pooling$ chamada de *max pooling*, a qual utiliza um $kernel$ de dimensões 2x2 para extrair apenas o maior valor enquadrado pela área $kernel$ (GÉRON, 2017).

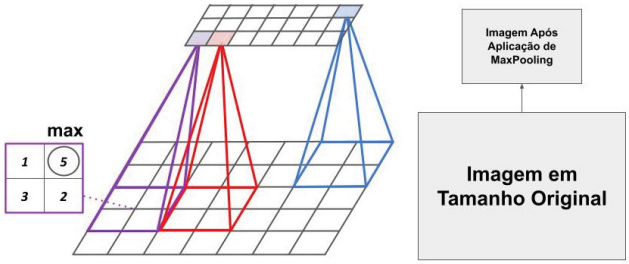


Figura 16. Camada *Max Pooling*. Fonte: Elaborado pelo autor.

Isto confere à camada de $pooling$ a habilidade de realizar um resumo estatístico da região em que seu $kernel$ é aplicado. Esta capacidade é o que permite às CNNs, por

exemplo, lidarem com entradas que variam de tamanho, e mesmo assim conseguirem detectar padrões já conhecidos (GOODFELLOW; BENGIO; COURVILLE, 2016).

Uma forma de visualizar todos os conceitos apresentados nesta seção é analisar um exemplo de arquitetura de uma CNN, como na Figura 17. Nela, é possível observar uma sequência de camadas de convolução e *pooling*, que extraem características das imagens de forma mais profunda à medida que avançam, e encolhem as imagens, respectivamente. Após a última camada de *pooling*, na camada *full connected*, os dados encontram-se vetorizados e podem ser utilizados por redes neurais, como LSTMs, BiLSTMs combinadas com funções de ativação (ReLU, Sigmóide, Softmax, etc.) para gerar predições, como a classificação de imagens ou textos, por exemplo (GÉRON, 2017).

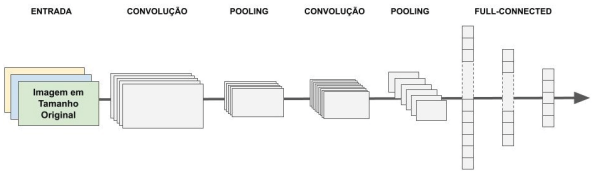


Figura 17. Exemplo de uma arquitetura CNN atrelada a uma rede de camadas *full connected*. Fonte: Elaborado pelo autor.

M. Word Embeddings: Word2Vec e FastText

O princípio de que "contextos similares tendem a possuir sentidos similares", conforme afirma Jurafsky e Martin (2014, pg. 96) diz respeito à hipótese de distribuição. Dele, surge o aprendizado por representação, o qual aprende representações dos sentidos das palavras, os chamados *word embeddings*, a partir da sua distribuição nos textos. Tais *embeddings* são representados por vetores semânticos, os quais permitem descrever as palavras e suas relações de vizinhança por meio de um espaço semântico multidimensional, como é apresentado na Figura 18 (JURAFSKY; MARTIN, 2014).

Figura 18. Espaço semântico multidimensional. Fonte: Elaborado pelo autor.

Neste sentido, a utilização deste tipo de representação permite calcular a similaridade entre palavras e sentenças dentro de um espaço, como na Figura 18. Nesta imagem é possível identificar a similaridade entre as palavras de acordo com a proximidade de sua distribuição no espaço, como é nitidamente retratado pela divisão dos três grupos de palavras. Desta forma, a aplicabilidade deste tipo de aprendizado por representação torna-se fundamental para tarefas envolvendo a compreensão de linguagem natural, como a atribuição de sentimentos a textos com base em sua similaridade (JURAFSKY; MARTIN, 2014). Afinal, o fato de que as representações são aprendidas de forma autônoma, torna desnecessária a morosa tarefa de extrair manualmente características dos textos (*feature engineering*) (BENGIO; COURVILLE; VINCENT, 2013).

Dentre os diversos modelos de vetores semânticos existentes, segundo Řehůřek (2021), o word2vec é um modelo que realiza o *embedding* de palavras em vetores de baixa dimensionalidade. Para isso, utiliza uma rede neural superficial composta por uma camada oculta, e dispõe de duas implementações distintas do modelo: o *Skip-gram* (SG) e o *Continuous-bag-of-words* (CBOW) (ŘEHůřEK, 2021).

O modelo SG recebe uma palavra de entrada e fornece uma predição de qual vizinhança de palavras ou contexto a palavra de entrada é mais próxima (ŘEHůřEK, 2021). Para realizar esta tarefa, o SG utiliza uma arquitetura que pode ser expressa através da equação:

$$Q = C \times (D + D \times \log_2(V)) \tag{20}$$

Em que C é a máxima distância entre as palavras; D são os vetores representativos e V o tamanho do vocabulário. Desta forma, cada palavra é utilizada como uma entrada de um classificador linear que utiliza *log* e uma camada de projeção contínua de dimensionalidade *N x D*. Na sequência, este classificador realiza a predição de palavras relacionadas a um intervalo pré-estabelecido de palavras que compõem o contexto de saída do modelo (MIKOLOV et al., 2013).

Já o modelo CBOW, ao invés de receber uma palavra de entrada, recebe um conjunto de palavras e prediz uma palavra central relacionada ao contexto informado (ŘEHůřEK, 2021). Sua arquitetura pode ser compreendida através da equação (MIKOLOV et al., 2013):

$$Q = N \times D + D \times \log_2(V) \tag{21}$$

Ambos os modelos e suas arquiteturas podem ser observados na Figura 19:

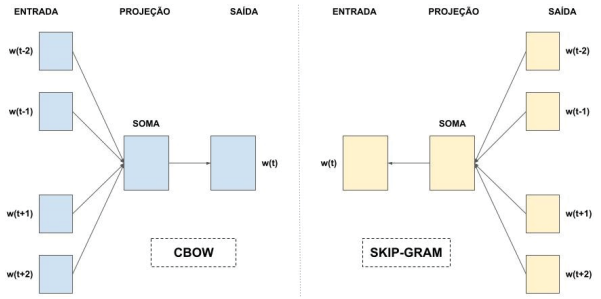


Figura 19. Modelos CBOW e Skip-gram. Fonte: Elaborado pelo autor.

Além do modelo word2vec, existem outros modelos que também geram *word embeddings*. Entre eles, o modelo Fast-

Text, proposto por Bojanowski et al. (2017), é uma extensão do word2vec. A característica principal deste modelo consiste em transformar cada palavra de um texto em múltiplos *n-grams* (sub-palavras). Então, para cada *n-gram* um $embedding$ do tipo Skip-gram é gerado, e cada palavra do texto é representada pela soma de todos os *embeddings* constituídos a partir dos *n-grams* (JURAFSKY; MARTIN, 2014).

N. Métricas de Desempenho

Modelos de classificação podem ser avaliados por meio da utilização de métricas que demonstram o seu desempenho diante de diferentes perspectivas. Dentre as possibilidades de avaliação existentes, destaca-se a matriz de confusão. Esta matriz apresenta o número de classificações corretas e incorretas para cada classe do $dataset$ em relação ao número total de classificações realizadas (GOLDSCHMIDT; PASSOS, 2005).

Como pode ser visualizado na Tabela I, a matriz apresenta o resultado em duas dimensões: as classes verdadeiras (o resultado correto esperado) e as classes preditas pelo classificador. Cada posição $M(C_i, C_j)$ diz respeito ao número de exemplos preditos na classe C_j que pertenciam à classe C_i, para k classes distintas (GOLDSCHMIDT; PASSOS, 2005).

Tabela I
MATRIZ DE CONFUSÃO PARA K CLASSES DISTINTAS

Classes	**Predita** C_1	**Predita** C_2	**...**	**Predita** C_k
Verdadeira C_1	M(C_1, C_1)	M(C_1, C_2)	...	M(C_1, C_k)
Verdadeira C_2	M(C_2, C_1)	M(C_2, C_2)	...	M(C_2, C_k)
...	...	...	...	...
Verdadeira C_k	M(C_k, C_1)	M(C_k, C_2)	...	M(C_k, C_k)

Neste sentido, uma outra perspectiva da matriz de confusão também útil para avaliação de um modelo de classificação pode ser observada na Figura 20. Nela, em cada posição da matriz são apresentadas as relações entre classes verdadeiras e classes preditas, conforme afirma Gébron (2017), em que:

- Verdadeiro positivo (VP): classe prevista como correta e que, de fato, era correta;
- Verdadeiro negativo (VN): classe prevista como errada, e que, de fato, era errada;
- Falso positivo (FP): classe prevista como correta, mas que era errada;
- Falso negativo (FN): classe prevista como errada, mas que era correta.

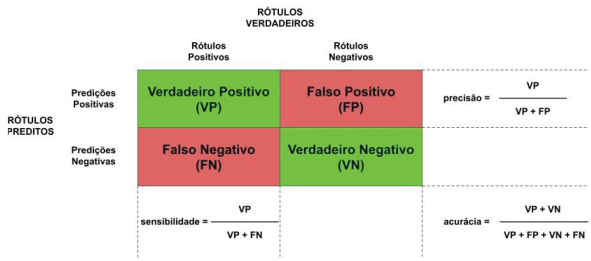


Figura 20. Matriz de Confusão binária. Fonte: Elaborado pelo autor.

Ainda em relação à Figura 20, é possível observar outras métricas de desempenho mais precisas e suas fórmulas, que podem ser utilizadas para avaliar um classificador (GÉRON, 2017), como as listadas nos tópicos a seguir, conforme Jurafsky e Martin (2014) definem:

- Acurácia ($accuracy$): é a porcentagem de classificações corretas realizadas pelo modelo diante de todas as observações apresentadas;
- Precisão ($precision$): é a porcentagem de classes atribuídas corretamente e que, de fato, são corretas;
- Sensibilidade ($recall$): é a porcentagem de observações que foram corretamente classificadas pelo modelo;
- F-1 *Score*: é uma métrica baseada na medida F, a qual representa uma média ponderada entre precisão e sensibilidade. Nesta métrica, o valor de β atua como um balanceador de pesos de importância entre precisão e sensibilidade. Neste sentido, quando $F_{\beta=1}$, estas duas métricas encontram-se balanceadas e, portanto, representam a medida F-1 $Score$. Tais métricas podem ser observadas nas equações seguintes:

$$F_\beta = \frac{\left(\beta^2 + 1\right) PR}{\beta^2 P + R}$$
$$\mathrm{F}_1 = \frac{2PR}{P + R} \tag{22}$$

O. CRISP-DM (Cross-Industry Standard Process for Data Mining)

Até o final da década de 90, enquanto a mineração de dados ganhava relevância, não existia um modelo de processamento de dados padrão que guiasse as empresas na obtenção dos melhores resultados em projetos desta área (SHEARER, 2000). O mercado carecia de uma descrição unificada de boas práticas que demonstrasse a viabilidade e o valor da mineração de dados como peça chave nos processos das empresas (CHAPMAN, 2000). Diante deste cenário, para resolver estes problemas, no final de 1996, um conjunto de empresas veteranas neste mercado criaram o *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Trata-se de uma metodologia não proprietária de mineração de dados (CHAPMAN, 2000). Segundo Miner et al. (2012), uma metodologia consiste num processo padronizado e documentado que guia a execução de atividades complexas por meio do uso de métodos, técnicas e ferramentas. A metodologia CRISP-DM fornece um guia completo que independe do tipo de contexto de negócio para criação de projetos da área de mineração de dados (SHEARER, 2000).

Desde a publicação destes estudos sobre a metodologia (CHAPMAN, 2000), (SHEARER, 2000); com o aumento da diversidade dos dados e das técnicas de exploração surgidas no período, Miner et al. (2012) e Martinez-Plumed et al. (2021) afirmam a partir de seus estudos que o CRISP-DM continua sendo uma das metodologias mais comuns para extração de conhecimento a partir de dados. Ao longo da sua trajetória, ela se desenvolveu e se tornou um dos modelos existentes de trajetória de ciência de dados, indicada para os casos nos quais

existe uma clareza dos objetivos de negócio e da mineração de dados no contexto. Na prática, essas trajetórias operam como um *template* flexível de planejamento de projetos envolvendo ciência de dados, os quais podem ser adaptados para as necessidades de cada cenário (MARTINEZ-PLUMED et al., 2021).

Embora seja inerentemente uma metodologia de mineração de dados, na prática, conforme afirma Miner et al. (2012, pg. 74-75), esta metodologia pode ser aplicada no contexto de mineração de textos. Afinal, tratam-se de áreas correlatas que se diferenciam primordialmente pelo tipo de dados utilizado no processo de descoberta de conhecimento (MINER et al., 2012). Respectivamente, a primeira envolvendo dados estruturados, e a outra dados semi ou não estruturados (textos). Diante deste cenário, a Figura 21 demonstra o ciclo de vida da metodologia CRISP-DM adaptado para mineração de textos, proposto por Miner et al. (2012, pg. 75), contendo 6 fases do processo de descoberta de conhecimento sobre textos.

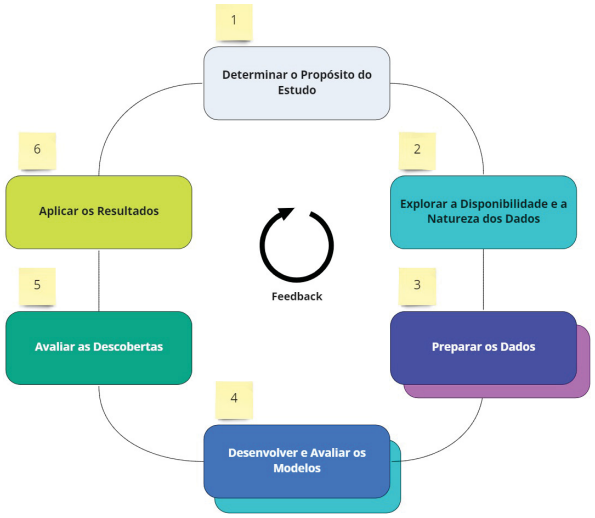


Figura 21. Fluxo de mineração de textos baseado na metodologia CRISP-DM. Fonte: Elaborado pelo autor.

1) Determinação do Propósito do Estudo: A primeira fase do CRISP-DM consiste no entendimento dos objetivos de negócio envolvidos no projeto. Afinal, conhecer do negócio permite compreender a motivação, as metas a serem alcançadas e, principalmente, como os dados se relacionam com o contexto. A partir deste entendimento, busca-se transformar os objetivos mapeados numa definição de problema de mineração de textos (SHEARER, 2000).

2) Exploração da Disponibilidade e Natureza dos Dados: Sequencialmente, a etapa de exploração parte da ideia de acessar e se familiarizar com os dados (CHAPMAN, 2000). Especificamente na mineração de textos, esta etapa consiste em identificar e entender como acessar as fontes de dados; coletar amostras e realizar as primeiras explorações nos dados para compreender sua natureza, disponibilidade, quantidade e qualidade em relação aos objetivos almejados (MINER et al., 2012).

3) Preparação dos dados: Após a exploração inicial dos dados, a etapa de preparação engloba todas as atividades que irão converter os dados brutos em dados estruturados. Seu objetivo consiste em preparar os dados para modelagem e extração de conhecimento. Na Figura 22 é possível visualizar as três atividades que compõem esta etapa de preparação, conforme descreve Miner et al. (2012, pg. 79), e que são descritas na sequência.

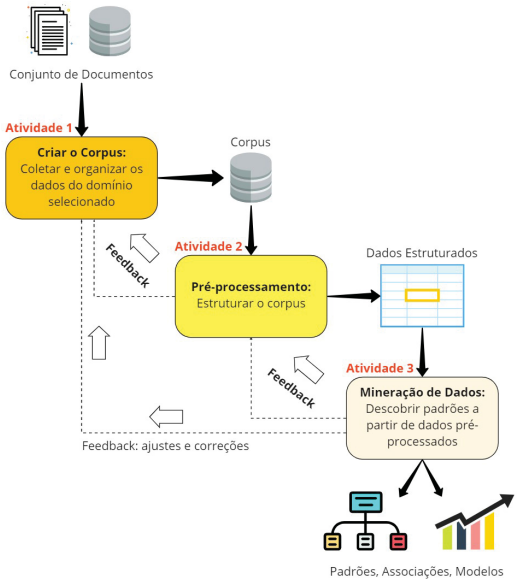


Figura 22. Fluxo das 3 atividades da etapa de preparação dos dados. Fonte: Elaborado pelo autor.

4) Estabelecer o Corpus: Na primeira atividade da etapa de preparação, busca-se coletar e organizar o *corpus* que atenderá os objetivos do projeto (MINER et al., 2012). Essencialmente, conforme afirma Martins et al. (2020), *corpus* ou *corpora* (plural), consistem, respectivamente, em um texto ou um conjunto de textos que servem como entrada de dados para processamento.

Normalmente, as atividades desta etapa distinguem textos, aqui chamados de documentos, em basicamente dois tipos: os que pertencem à web ou não. Algumas razões justificam esta distinção, como: (i) a ascensão da web e sua imensidão, que consagrou a área de mineração de textos elevando a disponibilidade de textos antes não alcançáveis; (ii) a forma e o estilo dos documentos web, normalmente escritos em *Hypertext Markup Language* (HTML), que diferem dos documentos em texto puro (MINER et al., 2012).

Para realizar a coleta dos dados provenientes de sites, Miner et al. (2012) sugere o desenvolvimento de um *web scraper* para realizar o procedimento de extração. Entretanto, a partir desta extração, os dados se encontram num estado "bruto", ou seja, distante do objetivo da mineração de texto de transformar dados semi ou não estruturados em um formato estruturado (MORAIS, 2007). Neste sentido, para avançar nestas transformações, segue-se para a etapa seguinte do processo, conhecida como pré-processamento.

5) Pré-processamento de dados: Com os dados separados e organizados, inicia-se a atividade 2, que consiste no pré-processamento do *corpus* para gerar uma representação estruturada dos dados, o formato esperado na entrada da etapa de mineração de dados (MINER et al., 2012). Como será aprofundado na subseção seguinte, para minerar os dados são aplicados algoritmos analíticos que, por natureza, são alimentados por representações numéricas (MARTINS et al., 2020). Desta maneira, os textos necessitam ser transformados em números para viabilizarem a compreensão e extração de conhecimento pelos algoritmos analíticos (MINER et al., 2012). Afinal, os computadores, em sua essência, são preparados para computar sequências numéricas que indicam a ausência ou passagem de corrente elétrica no *hardware*, ou seja, bem distante da linguagem natural representada pelos textos (MARTINS et al., 2020). Dadas estas condições e, para viabilizar a extração de conhecimento a partir dos textos, é que se utiliza o pré-processamento textual. Esta etapa consiste na aplicação de um grupo de transformações sobre os dados para representá-los em um formato estruturado (ARANHA, 2007).

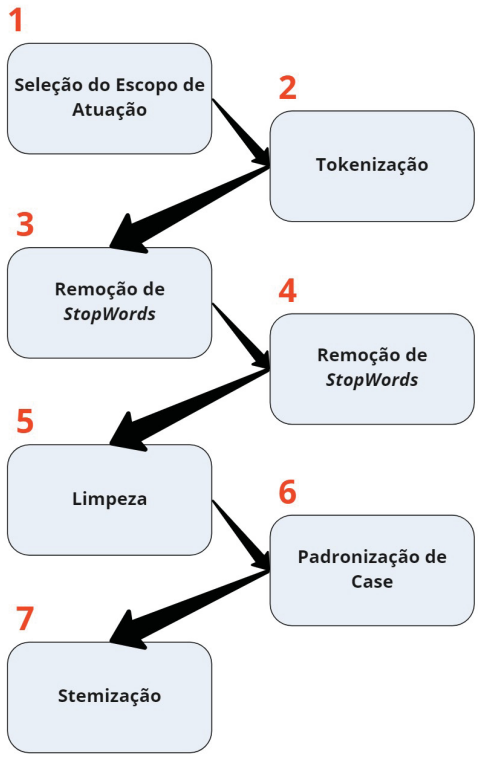


Figura 23. Sequência de tarefas de pré-processamento de texto aplicadas neste estudo. Fonte: Elaborado pelo autor.

Para cumprir estes requisitos, a etapa de pré-processamento aplica um conjunto de tarefas específicas denominadas como normalização de textos. A primeira tarefa, conhecida como *"tokenização"* consiste em isolar as unidades de um texto em *"tokens"* ou termos individuais, que podem ser palavras, frases, linhas e até mesmo parágrafos (MINER et al., 2012), utilizando como fronteira entre os termos, geralmente, o caractere em branco (ARANHA, 2007). Com os termos separados, já é possível partir para o próximo passo, conhecido como *"stopping"*, que consiste em remover *"stopwords"*, ou seja, palavras que não são relevantes para a compreensão do texto, como por exemplo os artigos a, as, o, os e entre outros termos. Além disso, também podem ser utilizadas regras de expressão regular, que consiste numa linguagem específica para especificar sequências de caracteres para realizar a limpeza de textos (JURAFSKY; MARTIN, 2014). Por sua vez, esta tarefa serve para filtrar caracteres especiais, pontuações, *tags* HTML e entre outros elementos indesejados (MARTINS et al., 2020). Uma boa prática consiste em nivelar o *case* das palavras em minúsculo ou maiúsculo, a fim de evitar termos com o mesmo significado com representações diferentes (JURAFSKY; MARTIN, 2014). Por fim, aplica-se a tarefa de *"stemming"*, que consiste na remoção de sufixos das palavras (ARANHA, 2007), como por exemplo, tratando as palavras *"walking"*, *"walks"*, *"walked"* e *"walker"* como *"walk"*. Com isto, o número de dimensões é reduzido por meio do agrupamento de palavras por conceitos. De modo geral, as tarefas apresentadas atuam para garantir um formato adequado para a vetorização, além de aprimorar a acurácia e a operação de algoritmos (MINER et al., 2012).

Após a sucessão de passos descritos, o resultado esperado da etapa de pré-processamento consiste em converter textos brutos em formato estruturado, como os vetores. Dado este resultado, os algoritmos de mineração de dados conseguem realizar análises e extrair resultados (MINER et al., 2012). Desta maneira, na subseção seguinte são apresentadas técnicas, tarefas e algoritmos utilizados para a mineração de dados a partir de textos.

6) Mineração de Dados ou Extração de Conhecimento: Por fim, conforme aponta a atividade 3, a fase de preparação de dados é encerrada com a extração de conhecimento, também conhecida como mineração de dados. Para isso, são aplicados os métodos de mineração de dados para o treinamento dos modelos de conhecimento a fim de gerar aprendizado sobre os dados textuais.

Por definição, a mineração de dados é uma área multidisciplinar e interdisciplinar que envolve inúmeros campos de conhecimento, entre eles a Inteligência Artificial e o Aprendizado de Máquina (CASTRO; FERRARI, 2016). Segundo Miner et al. (2012) esta área se intersecciona com a mineração de textos e fornece subsídios para diversas atividades, como a classificação e clusterização de textos e entre outras. Desta forma, a mineração de dados busca aplicar algoritmos para extrair conhecimentos a partir dos dados pré-processados (CASTRO; FERRARI, 2016). Estes algoritmos são baseados em técnicas que buscam explorar os dados para criar modelos de conhecimento que, por sua vez, possuem o objetivo de resolver um problema definido (GOLDSCHMIDT; PASSOS, 2005).

Para resolver estes utilizam-se algoritmos. Em sua essência, algoritmos são uma sequência de instruções que transformam um determinado tipo de dado em uma saída ou resultado (ALPAYDIN, 2016). Dentre as inúmeras técnicas e algoritmos dis-

poníveis na mineração de dados, conforme menciona Goldschmidt e Passos (2005), podem ser utilizados em problemas de extração de conhecimento os algoritmos disponibilizados por uma das subáreas da Inteligência Artificial (IA), a área de aprendizado de máquina (ou *machine learning*) (RUSSELL; NORVIG, 2010).

7) Avaliação dos modelos: Na penúltima fase, a de avaliação, todo o conhecimento extraído passa pela análise humana, afinal, é necessário avaliar se a construção do modelo atende os objetivos almejados no projeto (SHEARER, 2000). Assim, são avaliadas as métricas de qualidade do modelo, além de uma revisão de todo o processo percorrido para identificar se nenhuma etapa foi desconsiderada a fim de se evitar a propagação de erros (MINER et al., 2012). No final, desta fase, espera-se que o responsável pelo projeto decida se novas iterações devem ocorrer para melhorar os resultados dos modelos, ou se o projeto pode encerrar e prosseguir para a aplicação dos resultados em ambiente produtivo.

8) Publicação dos modelos: Na fase final, os resultados são organizados e disponibilizados para os usuários de acordo com o padrão acordado no projeto. Em muitos casos podem ser gerados desde simples relatórios ou páginas web para visualização, até mesmo a incorporação de modelos em ferramentas de apoio à tomada de decisão (CHAPMAN, 2000). Vale ressaltar, que os modelos podem perder sua acurácia ou relevância com o tempo, e podem ser necessárias novas iterações para mantê-los adequados aos novos dados utilizados no contexto de negócio (MINER et al., 2012).

III. Materiais e Métodos

Nesta seção são apresentadas as etapas propostas pela metodologia CRISP-DM para a criação do modelo de análise de sentimentos para avaliações do portal Glassdoor. Neste sentido, a seção encontra-se dividida e ordenada numa sequência evolutiva, que parte da extração e análise exploratória dos dados brutos em forma de texto, passando pelo pré-processamento, até a geração do modelo de aprendizado de máquina. A avaliação do modelo e os resultados obtidos são apresentados na seção IV.

A. Exploração e Análise de Dados

Para realização do estudo, foi realizada uma busca na Internet a fim de encontrar $datasets$ contendo textos de avaliações realizadas no site Glassdoor. Entretanto, naquele momento, constatou-se a inexistência de um $dataset$ específico que suprisse esta necessidade.

Desta forma, conforme sugere Miner et al. (2012), foi desenvolvido um *web scrapper* para extrair as avaliações do site. Os dados extraídos das avaliações podem ser visualizados na Tabela II.

Além disso, utilizou-se como base para a seleção das empresas listadas no $dataset$ os Rankings das Melhores Empresas de TI para se trabalhar no ano de 2020, segundo Gptw (2020b) e Gptw (2020c). Foram extraídas avaliações de 105 empresas, sendo 25 de grande porte e 80 de médio porte. Ao finalizar a extração das avaliações, o $dataset$ resultante atingiu a marca de 2.088.067 registros de avaliações únicas.

Tabela II
CAMPOS DO $DATASET$ GLASSDOOR

Id.	Coluna	Descrição
1	empresa	Nome da empresa
2	avaliacao	Nota atribuída à empresa num intervalo de [1, 5]
3	tempo_casa	Situação do colaborador e tempo de casa
4	data_comentario_cargo	Data da postagem da avaliação e cargo na empresa
5	titulo_avaliacao	Título da avaliação
6	pros_comentario	Texto descrevendo os pontos positivos da empresa
7	cons_comentario	Texto descrevendo os pontos negativos da empresa

A partir da primeira consolidação do $dataset$, seguiu-se para a etapa de exploração dos dados. Buscou-se analisar a distribuição das notas das avaliações, as quais serviriam como base para a separação das classes do $dataset$. Com base nesta análise, foi constatado que a distribuição das notas encontrava-se desbalanceada, como é possível visualizar na Figura 24.

Nesta figura, as barras representam cada uma das avaliações de 1 a 5 pelo número total de avaliações das notas distribuídas ao longo $dataset$. Desta maneira, como o desbalanceamento detectado pela análise exploratória poderia prejudicar a performance do classificador (GERÓN, 2017), outro ponto levantado com base na exploração dos dados, é que os textos que serão utilizados pela classificação (vide linhas 6 e 7 da Tabela II) encontravam-se separados. Com isso, foram adotados alguns tratamentos para corrigir tais deformidades apontadas a partir da análise na seção seguinte.

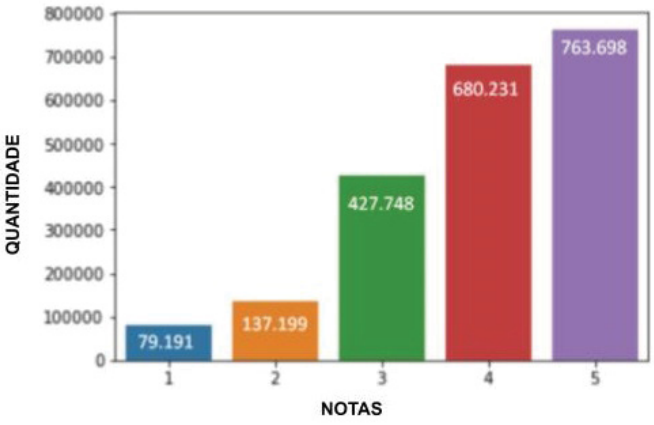


Figura 24. Número de avaliações por notas de 1 a 5. Fonte: Elaborado pelo autor.

B. Preparação dos Dados

Nesta etapa buscou-se pré-processar os dados a fim de que estivessem preparados para as etapas de treinamento dos modelos de aprendizado de máquina. Para isso, com base na

análise preliminar executada na etapa de exploração dos dados, foram realizados alguns tratamentos para correção de defeitos encontrados no $dataset$.

Num primeiro momento, buscou-se estabelecer a estratégia adotada para definir as classes do conjunto de dados. Afinal, como apresentado na Tabela II, o conjunto possuía apenas a nota como um possível rótulo, o qual não estaria em conformidade com o modelo ANP, que possuía as classes negativa, neutra e positiva para cada texto de avaliação. Desta forma, adotou-se a estratégia sugerida por (INDURKHYA; DUMERAU, 2010) também utilizada pela empresa ETI durante o treinamento do ANP, a fim de separar as notas 1 e 2 para representar a classe negativa; 3 a classe neutra; 4 e 5 a classe positiva.

Num segundo momento, buscou-se corrigir o desbalanceamento das notas. Para isso, foi estabelecida uma faixa de corte mínima para o número de avaliações de cada nota, fixado em 79.191 registros, conforme o menor número de avaliações da nota 1. Com isso, todas as notas passaram a possuir um número idêntico de número de registros para cada classe, como pode ser visualizado na Figura 25 (b).

Para chegar neste resultado, num primeiro momento foi realizado um embaralhamento do $dataset$ utilizando um $seed$ de número 10, a fim de manter a rastreabilidade das operações seguintes do estudo. Num segundo momento, seguindo a lógica da distribuição das notas pelas classes já comentada no início desta seção, optou-se por dividir as classes negativa e positiva pela metade, com 39.595 registros para cada nota, conforme demonstra a Figura 25 (a), a fim de balancear todas as classes com 79.190 registros. Por fim, para a classe neutra, composta apenas pela nota 3, foram mantidos os 79.190 registros.

Desta maneira, como afirmam Goldschmidt e Passos (2005), o balanceamento das classes previne o problema de prevalência de classes nos classificadores, em que os algoritmos de mineração de dados são influenciados por classes predominantes. Como resultado do balanceamento, obteve-se o número total de 237.750 registros, o qual foi utilizado como base para o treinamento das RNNs propostas neste estudo.

Após o balanceamento e separação das classes, o *corpus* utilizado como dado de entrada do modelo de classificação foi estabelecido. No caso deste estudo, um *corpus* corresponde à cada avaliação textual publicada e extraída do site Glassdoor. Desta forma, a fim de estabelecer um $corpus$ único para cada documento, realizou-se a concatenação das colunas de comentários positivos (linha 6) e negativos (linha 7) da Tabela II. A partir destes tratamentos preliminares realizados, formou-se um novo $dataset$ somente com os dados necessários para o treinamento do classificador, conforme demonstrado na Tabela III.

A partir da consolidação do $dataset$ apresentado na Tabela III, iniciou-se o pré-processamento do *corpus*. Este processo como um todo pode ser observado na Figura 26. Nele, iniciou-se com a transformação de todos os caracteres para forma minúscula, seguida da substituição de caracteres com acentuação para sua forma básica, como por exemplo: á para a, ó para o,

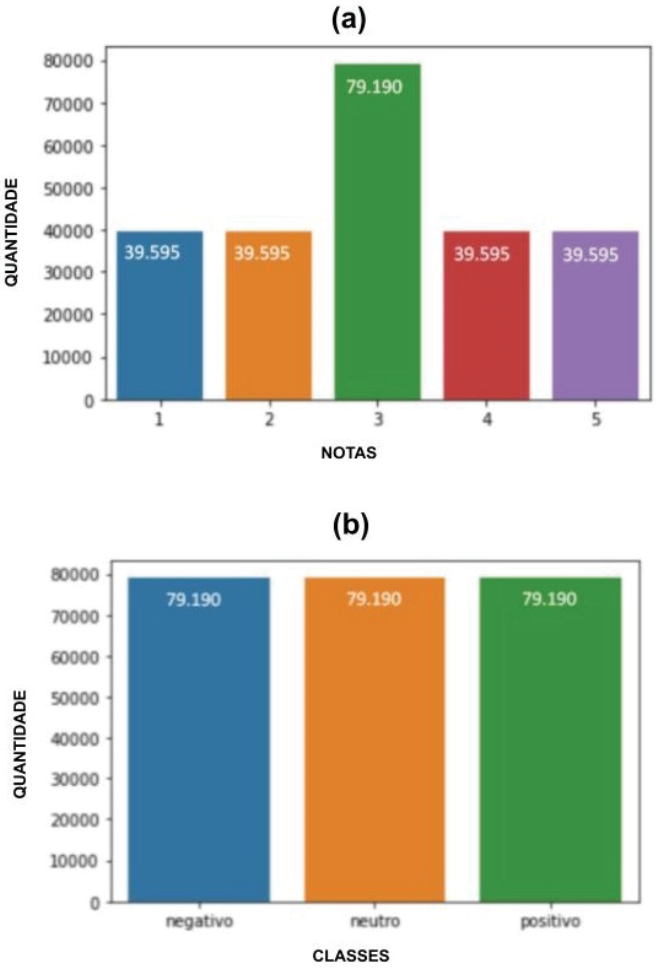


Figura 25. (a) Balanceamento de notas para formar as classes; (b) distribuição final do número de registros por classes no $dataset$.

Tabela III
VERSÃO PRÉ-PROCESSADA DO *DATASET* GLASSDOOR

Id.	Coluna	Descrição
1	texto_avaliacao	*Corpus* resultante da concatenação dos comentários positivos e negativos
2	classe	Uma classe exclusiva atribuída a cada documento, que pode ser: negativo ou neutro ou positivo

e assim por diante. Com isso, evitou-se que palavras idênticas fossem diferenciadas pela presença de caracteres maiúsculos, assim como a ausência ou adições de acentuações indevidas nas palavras, já que as avaliações partiam de textos livres escritos por usuários no Glassdoor.

Na sequência foram removidas as *stopwords*. Inclusive, as *stopwords* também foram transformadas em minúsculas e suas acentuações removidas a fim de estarem em conformidade com o *corpus* já pré-processado, evitando que *stopwords* não fossem detectadas. Após este passo, foram removidas as pontuações, caracteres especiais, *URL's*, números, espaços em

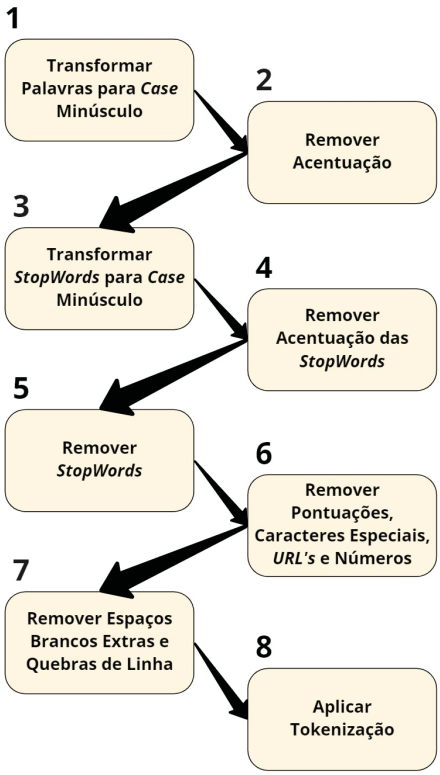


Figura 26. Sequência de passos aplicados no pré-processamento do $corpus$.

brancos extras entre as palavras e quebras de linha. Por fim, aplicou-se a técnica de tokenização, provida pela biblioteca Keras, para deixar o *corpus* preparado para a extração de características (TENSORFLOW, 2021).

C. Configuração de Ambiente

Os experimentos propostos neste estudo foram implementados utilizando o python (v.3.7.12) e diversas bibliotecas externas. Entre elas, cabe destacar a biblioteca Keras (v2.6.0), que fornece uma API para criação e manipulação de redes neurais e que foi utilizada para a implementação das redes LSTM, BiLSTM e CNN. Além disso, foram utilizadas as bibliotecas pandas (v1.1.5) numpy (v1.19.5) para manipulação de $dataframes$ e $arrays$, respectivamente; Tensorflow (v2.6.0) para habilitar o processamento dos modelos das TPUs; Gensim (v3.8.3) para treinamento do modelo word2vec; Scikit-learn (v.0.22.2) para realizar o $split$, *grid search* e *cross-validation* do $dataset$; Selenium (v3.141.0) e BeautifulSoup (v4.10.0) para o desenvolvimento do *web scrapper*; Matplotlib (v.3.2.2) para plotar gráficos para análise de dados; e Nltk (v3.2.5) para importação das *stopwords* na etapa de pré-processamento.

Toda a etapa de treinamento foi realizada no ambiente do Google Colaboratory, na versão Pro+, utilizando TPUs para processamento dos modelos. Optou-se pela utilização deste tipo de *hardware* após a realização de testes preliminares com CPUs e GPUs fornecidas pela plataforma, as quais apresentaram baixa performance em comparação com o desempenho das TPUs. Em específico ao ambiente do Google Colaboratory, vale ressaltar que a empresa fornecedora não especifica um tempo exato que o *hardware* fica alocado para o usuário, desta forma, longas execuções podem ser interrompidas a qualquer momento e inviabilizar treinamentos de modelos.

D. Estratégia de Treinamento

Para a condução do treinamento foi realizada a divisão do $dataset$ (237,570 registros) nas proporções 70/30, em que 70% dos dados (166,299 registros) foram direcionados para treinamento e 30% (71,271 registros) direcionados para testes. Além disso, foram utilizadas as técnicas de *cross-validation* e *grid search*. Na Tabela IV é apresentado o resultado de testes preliminares para cada combinação de parâmetro (candidato) da Figura 27 ao realizar o *grid search* em cada arquitetura proposta com *fold* = 10.

Vale ressaltar que estes testes preliminares foram interrompidos após 2 horas de execução, em função da inviabilidade do tempo estimado para a conclusão do treinamento de cada modelo e do presente estudo, como pode ser visto na Tabela IV. Desta forma, estes dados serviram como parâmetros para tomada de decisão do número de épocas, número de *folds* do *cross-validation* e a combinação de hiperparâmetros a serem utilizados no *grid search* para cada modelo.

Tabela IV
RESULTADOS DOS TESTES PRELIMINARES

Hiperparâmetros	**LSTM**	**BiLSTM**	**CNN-BiLSTM**
Número de Épocas	5	5	5
Tempo Médio de Treinamento Estimado por Candidato (em minutos)	14	28,5	54,5
Tempo Total Estimado de Treinamento do *Grid Search* (em dias)	52,5	136,8	204,3
Acurácia Obtida em Treinamento	0,978	0,907	0,982
Acurácia Obtida em Teste	0,979	0,945	0,988
Taxa de Perda Obtida em Teste	0,110	0,1583	0,087

```
# define the grid search parameters
batch_size = [16, 32, 64, 128]
epochs = [5, 10, 20]
embedding_dim = [16, 32, 64, 128, 512]
lstm_units = [16, 32, 64, 128, 512]
activation = ['softmax', 'relu']
optimizer = ['Adam', 'RMSProp', 'Adagrad']
dropout_rate = [0.2, 0.4, 0.5]
```

Figura 27. Combinação de hiperparâmetros utilizada nos testes preliminares.

Na Tabela IV nota-se que conforme a arquitetura ganha complexidade, o tempo estimado de treinamento aumenta.

Além disso, para todas as arquiteturas o número de 5 épocas foi capaz de demonstrar resultados satisfatórios de acurácia com a base de testes.

Desta forma, para realizar o treinamento de todos os modelos, os fatores de continuidade da disponibilidade dos ambientes e tempo de treinamento dos modelos foram determinantes. Afinal, como já mencionado, o ambiente de execução do Google Colaboratory não especifica um tempo exato para execuções contínuas de TPUs, o que chegou a causar interrupções inesperadas durante outras execuções antecedentes. Além disso, embora fosse ideal realizar o *grid search* com todas as combinações apresentadas na Figura 27, a explosão combinatória de aproximadamente 6912 candidatos multiplicado por 10 $folds$ resultaria em 69120 treinamentos, levando em conta o tempo estimado para cada candidato de cada arquitetura.

Diante da inviabilidade descrita neste cenário, optou-se por utilizar para todos os treinamentos deste estudo o número de 5 épocas em função dos bons resultados alcançados; 5 $folds$ por ser o número padrão indicado pela biblioteca Keras, além de diminuir a explosão combinatória pela metade e ainda manter o *cross-validation*; assim como a seleção dos hiperparâmetros mais relevantes para o *grid search* conforme as justificativas apresentadas nas tabelas de cada arquitetura na subseção III-F.

E. Word2Vec e FastText

Com o *corpus* tokenizado aplicou-se o word2vec a fim de gerar os *word embeddings* e extrair as relações semânticas das palavras contidas nas avaliações. Este tipo de modelo de vetor semântico foi escolhido em função da sua presença em outros trabalhos relacionados à classificação de textos utilizando LSTMs (RHANOUI et al., 2019) (ARORA; KHODAK; SAUNSHI, 2018) (RAO; SPASOJEVIC, 2016).

Para realizar o treinamento do modelo word2vec foram utilizados os hiperparâmetros, valores e justificativas para configuração do modelo apresentados na Tabela V. No total, foram gerados 4 modelos word2vec distintos, de acordo com cada tamanho (*size*) apresentado na Tabela V, mantendo os demais parâmetros em valores padrões ou de acordo com os valores apresentados na tabela. Por fim, a arquitetura word2vec selecionada para o estudo foi a CBOW, conforme sua aplicação em um trabalho similar de classificação de textos em Li et al. (2018).

Já o modelo ANP foi treinado a partir da utilização do FastText, sendo esta uma estratégia originalmente proposta pela ETI. Na Tabela VI podem ser observados os parâmetros e valores propostos para o modelo FastText.

F. Arquiteturas das Redes LSTM, BiLSTM e CNN-BiLSTM

Para atender os objetivos do estudo foram criadas três redes neurais recorrentes com arquiteturas distintas, a fim de avaliar qual modelo apresentou a melhor acurácia em predições de avaliações do Glassdoor. Todas as arquiteturas implementadas neste estudo foram baseadas em trabalhos relacionados à classificação de textos utilizando RNNs, como listados na Tabela VII. Nesta tabela também foi incluído o modelo ANP, cuja arquitetura e hiperparâmetros propostos pela ETI também são apresentados nesta subsecção.

Tabela V
HIPERPARÂMETROS DO MODELO WORD2VEC

Hiperparâmetro	Valor(es)	Justificativa
$size$	[32, 64, 128, 512]	Conforme os valores de *embedding layers size* sugeridos por Rao e Spasojevic (2016)
$window$	5	Conforme exemplo disponível na documentação do word2vec (ŘEHůřEK, 2021) e na afirmação de Mikolov et al. (2013, pg. 4) que diz que "o aumento do intervalo melhora a qualidade dos vetores de palavras resultantes"
min_count	1	De acordo com o exemplo disponível na documentação do word2vec (ŘEHůřEK, 2021) e na sugestão de Sachan, Zaheer e Salakhutdinov (2020) para aproveitar o máximo do vocabulário a fim de não desperdiçar palavras que possuem alta representatividade para a determinação de uma classe

Tabela VI
HIPERPARÂMETROS DO MODELO FASTTEXT

Hiperparâmetro	Valor(es)
$size$	200
$window$	1
$total\_examples$	73,634
$epochs$	10

Tabela VII
MODELOS DE REDES NEURAIS PROPOSTOS NO ESTUDO E AS REFERÊNCIAS DE SUAS ARQUITETURAS

Id.	Arquitetura	Referência
1	LSTM	(RAO; SPASOJEVIC, 2016)
2	BiLSTM	(SACHAN; ZAHEER; SALAKHUTDINOV, 2020)
3	CNN-BiLSTM	(RHANOUI et al., 2019)
4	CNN (Modelo ANP)	Modelo original proposto pela ETI

1) LSTM - Arquitetura e Hiperparâmetros: A decisão de trabalhar com LSTMs partiu da sua especialização em dados sequenciais e sua eficácia comprovada em problemas envolvendo processamento de linguagem natural, como classificação de textos (JURFSKY; MARTIN, 2014). Parte disso, também deve-se a sua capacidade de capturar e manter dependências em dados sequenciais longos, como aponta (ZHOU et al., 2016). Na Figura 28 é apresentada a arquitetura LSTM proposta por Rao e Spasojevic (2016).

Para realizar o treinamento deste modelo foram selecionados alguns hiperparâmetros para a configuração da rede LSTM. Na Tabela VIII, são listadas as camadas e seus respectivos hiperparâmetros, se foram incluídos no *grid search*, além de seus valores e devidas justificativas.

2) BiLSTM - Arquitetura e Hiperparâmetros: A segunda arquitetura proposta neste estudo se baseia na utilização de

Tabela VIII
HIPERPARÂMETROS UTILIZADOS NA ARQUITETURA LSTM

Camada	**Hiperparâmetro**	**Inclusão no *Grid Search*?**	**Valor(es)**	**Justificativa**
Embedding	*input_dim*	Não	32,282	Tamanho do vocabulário extraído do *dataset*. Conforme afirma Sachan, Zaheer e Salakhutdinov (2020), a não limitação de palavras frequentes do vocabulário evita a perda de palavras que possam ter grande representatividade para as classes
Embedding	*output_dim*	Sim	[32, 64, 128, 512]	Valores sugeridos em trabalhos relacionados a Sachan, Zaheer e Salakhutdinov (2020) e Rao e Spasojevic (2016)
Embedding	*weights*	Não	Matriz de pesos do modelo word2vec	Conforme sugerido no trabalho de Sachan, Zaheer e Salakhutdinov (2020)
Embedding	*trainable*	Não	*True*	Para permitir o treinamento dos pesos ao longo da rede. Esta estratégia também foi proposta pela ETI para o treinamento do modelo ANP
Embedding	*input_length*	Não	230	Média aproximada do tamanho de todos os textos de avaliação do *dataset*
LSTM	*units*	Sim	[32, 64, 128, 512]	Valores pareados aos do hiperparâmetro *output_dim*, pois, conforme afirma Rao e Spasojevic (2016), o classificador tende a obter melhor acurácia quando o número de unidades LSTM é igual ao *output_dim* da camada de *Embedding*
Dropout	*rate*	Não	0.5	Valor proposto em trabalhos relacionados (SACHAN; ZAHEER; SALAKHUTDINOV, 2020), (ZHOU; SUN; LIU; LAU, 2015)
Dense	*units*	Não	3	Valor correspondente ao número de classes do *dataset*
Dense	*activation*	Não	Softmax	Conforme mencionam Russel e Norvig (2010) e Jurafksy e Martin (2014), a função softmax é geralmente utilizada na última camada da rede neural
-	*loss*	Não	*Categorical_ crossentropy*	Entrada do *array* de classes na rede em formato *one_hot*
-	*optimizer*	Não	Adam	Obteve melhor performance no problema de classificação de textos no trabalho de Rao e Spasojevic (2016)
-	*batch_size*	Sim	[32, 64, 128]	Valores 32 e 64 sugeridos no estudo de Rhanoui et al. (2019). Valor 128 selecionado a partir da lógica de seleção de números em base 10 (GÉRON, 2017) e da afirmação de Rao e Spasojevic (2016) que números maiores tendem a convergir mais rápido

Tabela IX
HIPERPARÂMETROS UTILIZADOS NA ARQUITETURA BILSTM

Camada	**Hiperparâmetro**	**Inclusão no *Grid Search*?**	**Valor(es)**	**Justificativa**
Embedding	*input_dim*	Não	32,282	Tamanho do vocabulário extraído do *dataset*. Conforme afirma Sachan, Zaheer e Salakhutdinov (2020), a não limitação de palavras frequentes do vocabulário evita a perda de palavras que possam ter grande representatividade para as classes
Embedding	*output_dim*	Sim	[128, 300]	O valor 128 possuiu melhor acurácia na arquitetura LSTM (RAO; SPASOJEVIC, 2016), enquanto o valor 300 foi sugerido no estudo da arquitetura BiLSTM (SACHAN; ZAHEER; SALAKHUTDINOV, 2020)
Embedding	*weights*	Não	Matriz de pesos do modelo word2vec	Conforme sugerido no trabalho de Sachan, Zaheer e Salakhutdinov (2020)
Embedding	*trainable*	Não	*True*	Para permitir o treinamento dos pesos ao longo da rede. Esta estratégia também foi proposta pela ETI para o treinamento do modelo ANP
Embedding	*input_length*	Não	230	Média aproximada do tamanho de todos os textos de avaliação do *dataset*
Dropout	*rate*	Não	0.5	Valor proposto na arquitetura BiLSTM original (SACHAN; ZAHEER; SALAKHUTDINOV, 2020)
BiLSTM	*units*	Sim	[32, 512]	O valor 32 tende a possuir melhor acurácia quando combinado com valor 128 de *output_dim* da camada *Embedding* na arquitetura LSTM (RAO; SPASOJEVIC, 2016). O valor 512 foi sugerido no estudo da arquitetura BiLSTM (SACHAN; ZAHEER; SALAKHUTDINOV, 2020)
BiLSTM	*dropout*	Não	0.5	Valor proposto na arquitetura BiLSTM original (SACHAN; ZAHEER; SALAKHUTDINOV, 2020)
Dense	*units*	Não	3	Valor correspondente ao número de classes do *dataset*
Dense	*activation*	Não	Softmax	Conforme mencionam Russel e Norvig (2010) e Jurafksy e Martin (2014), a função softmax é geralmente utilizada na última camada da rede neural
-	*loss*	Não	*Categorical_ crossentropy*	Entrada do *array* de classes na rede em formato *one_hot*
-	*optimizer*	Não	Adam	Obteve melhor performance no problema de classificação de textos no trabalho de Rao e Spasojevic (2016)
-	*batch_size*	Sim	[32, 64]	Valores 32 e 64 sugeridos nos estudos de Rhanoui et al. (2019)

BiLSTM. Sua escolha parte da premissa do aproveitamento das vantagens propostas pela LSTM. Outro fator, é de que devido sua arquitetura ser baseada em uma BRNN, seja possível aproveitar melhor o contexto dos textos e obter melhores resultados de predição (GRAVES; FERNANDEZ; SCHMIDHUBER, 2005).

Na Figura 29 é possível analisar a arquitetura BiLSTM implementada que foi proposta por Sachan, Zaheer e Salakhutdinov (2020). Enquanto na Tabela IX, são listadas as camadas e seus respectivos hiperparâmetros, se foram incluídos no *grid search*, além de seus valores e devidas justificativas.

3) CNN-BiLSTM - Arquitetura e Hiperparâmetros: A terceira arquitetura proposta é um modelo híbrido de CNN com BiLSTM. A aposta na utilização de CNNs para tarefas envolvendo o processamento de linguagem, deve-se à sua capacidade de identificar padrões através do espaço, assim como detectar padrões locais e suportar a variância de locais (MINAEE et al., 2021). Neste sentido, alguns trabalhos na literatura já utilizaram CNNs para classificação de textos (KIM; JEONG, 2019) (LI et al., 2018).

O propósito do modelo híbrido entre CNN e BiLSTM é combinar os pontos fortes da CNN com a especialidade das LSTMs em trabalhar com dados sequenciais longos e capturar dependências entre palavras e estruturas (MINAEE et al., 2021).

Na Figura 30 é possível analisar a arquitetura CNN-BiLSTM proposta por Rhanoui et al. (2019) e, na Tabela X, são listadas as camadas e seus respectivos hiperparâmetros, se foram incluídos no *grid search*, além de seus valores e devidas justificativas.

4) CNN (Modelo ANP) - Arquitetura e Hiperparâmetros: Por fim, o último modelo utilizado como base para os estudos deste artigo é o modelo ANP. Fornecido pela empresa ETI, este modelo foi criado originalmente com uma arquitetura CNN, a qual pode ser observada na Figura 31. Já na Tabela XI, é possível analisar os hiperparâmetros utilizados durante o treinamento.

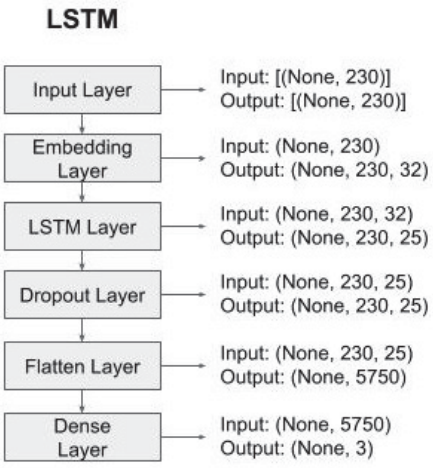


Figura 28. Arquitetura LSTM.

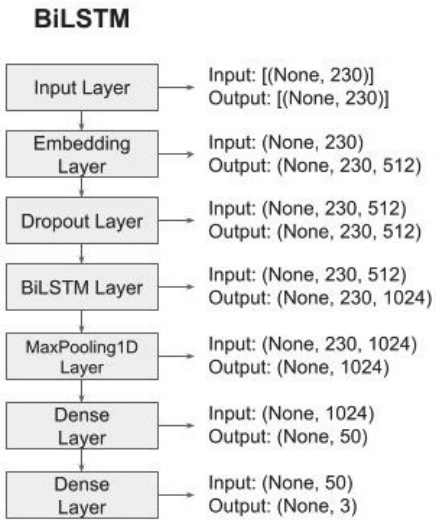


Figura 29. Arquitetura BiLSTM.

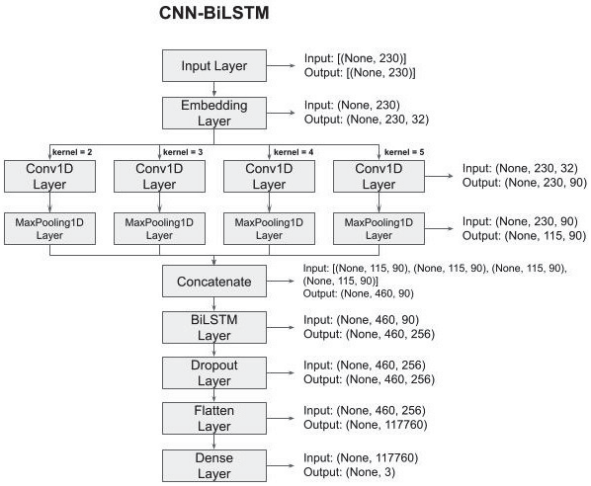


Figura 30. Arquitetura CNN-BiLSTM.

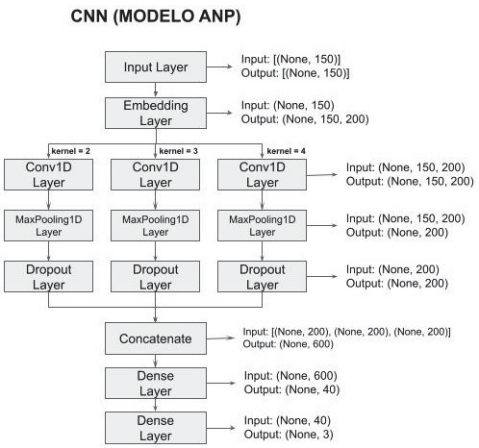


Figura 31. Arquitetura CNN implementada originalmente pela empresa ETI, referente ao modelo ANP.

Tabela X
HIPERPARÂMETROS UTILIZADOS NA ARQUITETURA CNN-BILSTM

Camada	**Hiperparâmetro**	**Inclusão no *Grid Search*?**	**Valor(es)**	**Justificativa**
Embedding	*input_dim*	Não	32,282	Tamanho do vocabulário extraído do *dataset*. Conforme afirma Sachan, Zaheer e Salakhutdinov (2020), a não limitação de palavras frequentes do vocabulário evita a perda de palavras que possam ter grande representatividade para as classes
Embedding	*output_dim*	Sim	[128, 300]	O valor 128 possuiu melhor acurácia na arquitetura LSTM (RAO; SPASOJEVIC, 2016), enquanto o valor 300 foi sugerido no estudo da arquitetura BiLSTM (SACHAN; ZAHEER; SALAKHUTDINOV, 2020)
Embedding	*weights*	Não	Matriz de pesos do modelo word2vec	Conforme sugerido no trabalho de Sachan, Zaheer e Salakhutdinov (2020)
Embedding	*trainable*	Não	*True*	Para permitir o treinamento dos pesos ao longo da rede. Esta estratégia também foi proposta pela ETI para o treinamento do modelo ANP
Embedding	*input_length*	Não	230	Média aproximada do tamanho de todos os textos de avaliação do *dataset*
Convolution	*filters*	Não	90	Valor utilizado na arquitetura original CNN-BiLSTM (RHANOUI et al., 2019)
Convolution	*kernel_size*	Não	[2, 3, 4, 5]	Cada valor foi utilizado exclusivamente em uma das 4 *convolution layers* da arquitetura. Valores utilizados na arquitetura original CNN-BiLSTM (RHANOUI et al., 2019)
Convolution	*activation*	Não	ReLU	Função de ativação proposta na arquitetura original CNN-BiLSTM (RHANOUI et al., 2019)
MaxPooling	*pool_size*	Não	2	Valor utilizado na arquitetura original CNN-BiLSTM (RHANOUI et al., 2019)
BiLSTM	*units*	Sim	[32, 512]	O valor 32 tende a possuir melhor acurácia quando combinado com valor 128 de *output_dim* da camada *Embedding* na arquitetura LSTM (RAO; SPASOJEVIC, 2016). Enquanto o valor 512 foi sugerido no estudo da arquitetura BiLSTM (SACHAN; ZAHEER; SALAKHUTDINOV, 2020)
Dropout	*rate*	Não	0.5	Valor proposto em trabalhos relacionados (SACHAN; ZAHEER; SALAKHUTDINOV, 2020), (ZHOU et al., 2015)
Dense	*units*	Não	3	Valor correspondente ao número de classes do *dataset*
Dense	*activation*	Não	Softmax	Conforme mencionam Russel e Norvig (2010) e Jurafksy e Martin (2014), a função softmax é geralmente utilizada na última camada da rede neural
-	*loss*	Não	*Categorical_ crossentropy*	Entrada do *array* de classes na rede em formato *one_hot*
-	*optimizer*	Não	Adam	Obteve melhor performance no problema de classificação de textos no trabalho de Rao e Spasojevic (2016)
-	*batch_size*	Sim	[32, 64]	Valores 32 e 64 sugeridos no estudo de Rhanoui et al. (2019)

Tabela XI
HIPERPARÂMETROS UTILIZADOS NA ARQUITETURA CNN (MODELO ANP)

Camada	**Hiperparâmetro**	**Valor(es)**
Input	*shape*	(150,)
Embedding	*input_dim*	160,990
Embedding	*output_dim*	200
Embedding	*weights*	Matriz de pesos do modelo pré-treinado FastText
Embedding	*trainable*	*True*
Convolution	*filters*	200
Convolution	*kernel_size*	[2, 3, 4]
Convolution	*padding*	*same*
Convolution	*activation*	ReLU
Convolution	*kernel_regularizer*	L2(0.001)
Dropout	*rate*	0.1
Dense	*units*	40 (camada oculta) e 3 (última camada)
Dense	*activation*	ReLU (camada oculta) e Softmax (última camada)
-	*loss*	*categorical_crossentropy*
-	*optimizer*	Adam
-	*batch_size*	128

IV. Resultados e Discussões

Nesta seção são apresentados os resultados obtidos com os treinamentos e testes dos modelos propostos neste estudo um resumo geral destes resultados, o qual demonstra que o modelo CNN-BiLSTM foi capaz de superar os demais modelos propostos. Assim como, o modelo ANP demonstrou baixa eficácia ao realizar predições com o *dataset* Glassdoor.

Na sequência, são expostos os resultados individuais de cada modelo contendo um relatório de classificação, a matriz de confusão e os gráficos de acurácia e perda. Exceto para o modelo ANP, cujos resultados são apresentados somente através do relatório de classificação e matriz de confusão a fim de demonstrar a capacidade do modelo em realizar predições com o *dataset* Glassdoor. No final da seção, são realizadas as devidas discussões sobre as descobertas e resultados obtidos.

Tabela XII
RESUMO: RESULTADO FINAL DA PREDIÇÃO DOS MODELOS

Arquitetura	***Batch Size***	***Embedding Units***	**LSTM *Units***	**Acurácia em Testes**
LSTM	32	512	128	0,9883
BiLSTM	32	300	512	0,9847
CNN-BiLSTM	32	128	32	**0,9896**
CNN (Modelo ANP)	128	150	-	0,3350

A. LSTM

A seguir são apresentados os resultados obtidos com predições realizadas sobre o conjunto de testes do modelo LSTM, que podem ser observados no relatório classificação contendo métricas específicas para avaliação do modelo (vide Figura 32), matriz de confusão (vide Figura 33) e gráficos de acurácia (Figura 34) e perda (Figura 35).

B. BiLSTM

Os resultados obtidos com o treinamento do modelo BiLSTM e a sua eficácia com predições em conjunto de testes podem ser observados no seu relatório de classificação (Figura 36), matriz de confusão (Figura 37) e gráficos de acurácia (Figura 38) e perda (Figura 39).

C. CNN-BiLSTM

Na sequência podem ser observados os resultados obtidos com o modelo CNN-BiLSTM a partir de predições no conjunto de testes, conforme o relatório de classificação (Figura 40), matriz de confusão (Figura 41) e gráficos de acurácia (Figura 42) e perda (Figura 43).

D. CNN (Modelo ANP)

O modelo ANP, como já mencionado, foi re-treinado neste estudo e sua eficácia colocada à prova diante de todo *dataset* do Glassdoor. Nas Figuras 44 e 45, são apresentados o relatório de classificação do modelo e sua matriz de confusão, respectivamente.

	precision	recall	f1-score	support
negativo	0.9938	0.9963	0.9950	23737
neutro	0.9819	0.9898	0.9858	23665
positivo	0.9893	0.9789	0.9841	23869
accuracy			0.9883	71271
macro avg	0.9883	0.9883	0.9883	71271
weighted avg	0.9883	0.9883	0.9883	71271

Figura 32. Relatório de classificação do modelo LSTM.

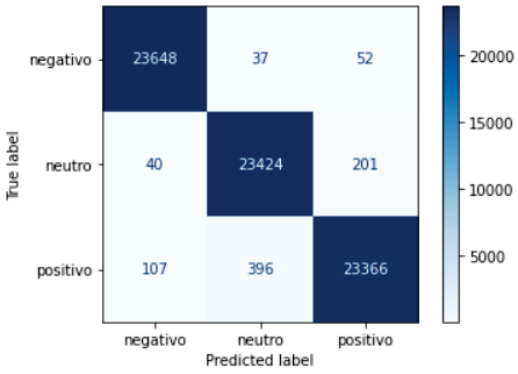


Figura 33. Matriz de confusão do modelo LSTM.

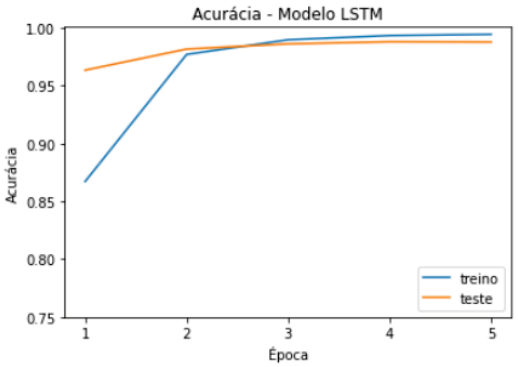


Figura 34. Gráfico de acurácia do modelo LSTM.

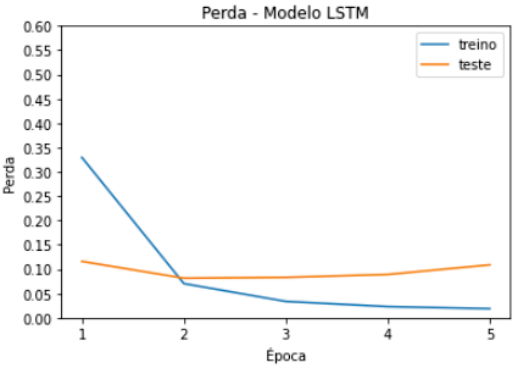


Figura 35. Gráfico de perda do modelo LSTM.

	precision	recall	f1-score	support
negativo	0.9889	0.9970	0.9929	23737
neutro	0.9749	0.9894	0.9821	23665
positivo	0.9904	0.9678	0.9790	23869
accuracy			0.9847	71271
macro avg	0.9847	0.9847	0.9847	71271
weighted avg	0.9848	0.9847	0.9847	71271

Figura 36. Relatório de classificação do modelo BiLSTM.

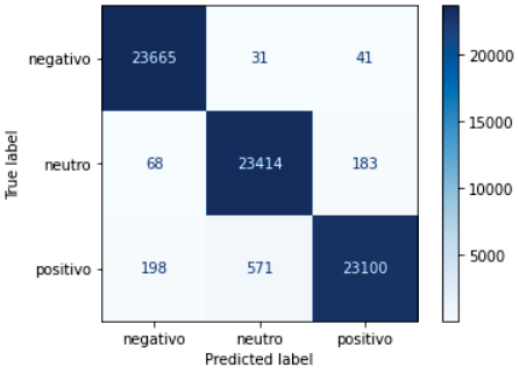


Figura 37. Matriz de confusão do modelo BiLSTM.

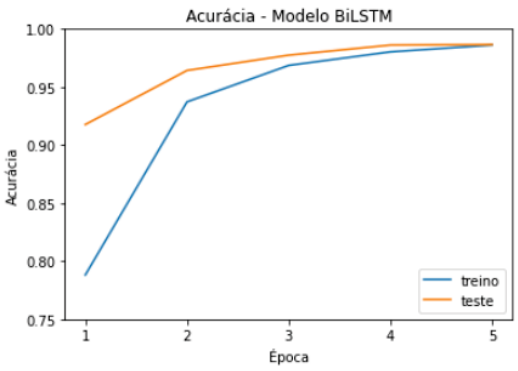


Figura 38. Gráfico de acurácia do modelo BiLSTM.

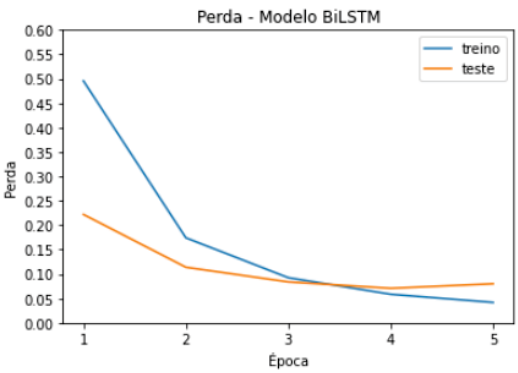


Figura 39. Gráfico de perda do modelo BiLSTM.

	precision	recall	f1-score	support
negativo	0.9944	0.9956	0.9950	23737
neutro	0.9892	0.9858	0.9875	23665
positivo	0.9852	0.9873	0.9862	23869
accuracy			0.9896	71271
macro avg	0.9896	0.9896	0.9896	71271
weighted avg	0.9896	0.9896	0.9896	71271

Figura 40. Relatório de classificação do modelo CNN-BiLSTM.

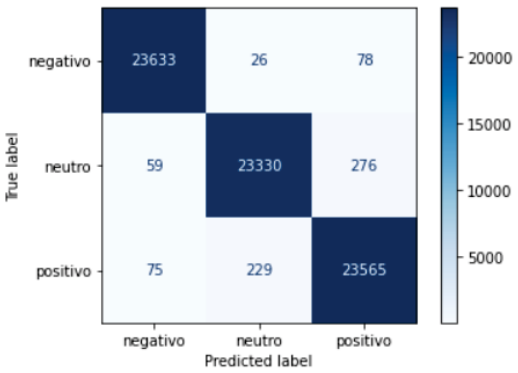


Figura 41. Matriz de confusão do modelo CNN-BiLSTM.

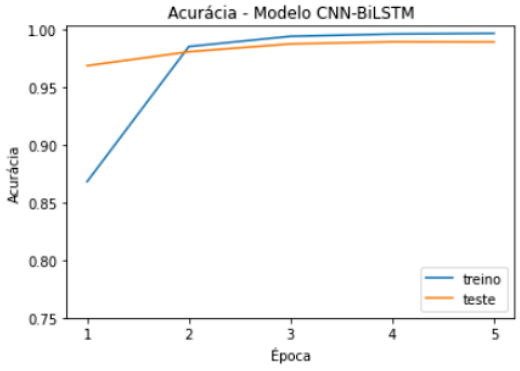


Figura 42. Gráfico de acurácia do modelo CNN-BiLSTM.

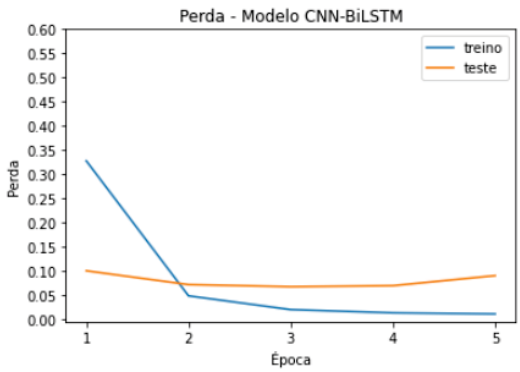


Figura 43. Gráfico de perda do modelo CNN-BiLSTM.

	precision	recall	f1-score	support
negativo	0.3412	0.2885	0.3126	79190
neutro	0.3198	0.0560	0.0953	79190
positivo	0.3338	0.6607	0.4435	79190
accuracy			0.3350	237570
macro avg	0.3316	0.3350	0.2838	237570
weighted avg	0.3316	0.3350	0.2838	237570

Figura 44. Relatório de classificação do modelo ANP no $dataset$ Glassdoor.

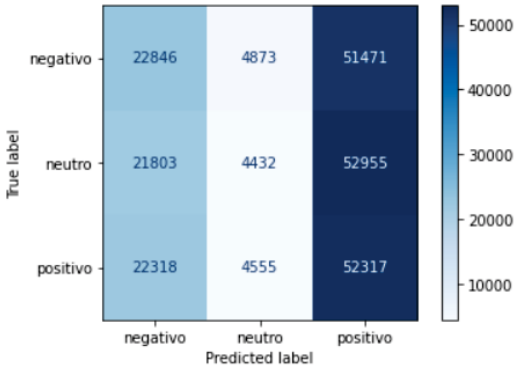


Figura 45. Matriz de confusão do modelo ANP.

E. Discussões

Como pode-se observar, a maior acurácia obtida (0,9896) com os experimentos foi alcançada pela arquitetura CNN-BiLSTM, cujo resultado já era uma tendência desde os testes preliminares apresentados no artigo (vide Tabela IV). No trabalho realizado por Rhanoui et al. (2019), o qual inspirou a arquitetura CNN-BiLSTM empregada neste estudo, um resultado similar foi obtido ao comparar o desempenho dos modelos LSTM, BiLSTM e CNN-BiLSTM no contexto de análise de sentimento em textos. A hipótese apresentada pelo autor Rhanoui et al. (2019) para justificar o resultado, e que pode ser considerada neste estudo, é de que a união das capacidades de extração de features da CNN e a especialidade da BiLSTM em aprender de forma bidirecional e manter dependências entre palavras por longos períodos, contribuiu para o melhor resultado.

Além disso, vale mencionar que naquele estudo o modelo CNN-BiLSTM alcançou a acurácia de 0,9066, seguido pelas acurácias dos modelos BiLSTM (0,8640) e LSTM (0,8587) (RHANOUI et al., 2019). Entretanto, neste presente estudo, a diferença entre a acurácia entre os modelos foi mínima. Assim como, o modelo LSTM ficou em segundo lugar, superando a acurácia da sua concorrente BiLSTM, que deveria apresentar melhores resultados em função da sua capacidade de aprender características de forma bidirecional. Outro fato interessante, é que a LSTM apresentou sua melhor acurácia com um número maior de *embedding units* (512), sendo esta relação já apontada por Rao e Spasojevic (2016) em seu estudo. Quanto ao $batch\_size$, todos os modelos apresentaram boa performance com o menor número de unidades (32), resultado que também foi obtido com o estudo de Rao e Spasojevic (2016). Entretanto, somente a observação do $batch\_size$ parece fazer sentido para os resultados obtidos, afinal a tendência observada é de que sejam necessárias menores quantidades de *embedding units* à medida em que a complexidade da arquitetura aumenta (ver Tabela XII).

De forma geral, os modelos apresentaram equilíbrio ao realizar a análise de sentimento de três classes diferentes, o que pode ser um reflexo do balanceamento realizado na etapa de pré-processamento do $dataset$. Outro ponto a ser mencionado é que o número de 5 épocas proposto nos testes preliminares foram suficientes para convergir os modelos propostos. Neste sentido, mesmo com acurácias elevadas, os modelos foram capazes de generalizar ao serem expostos ao conjunto de testes.

Por fim, o modelo ANP apresentou baixa acurácia para predições com o $dataset$ do Glassdoor, confirmando a suspeita da ETI a respeito da baixa acurácia do modelo ANP em classificar dados de um contexto textual diferente do utilizado em seu treinamento. Entretanto, vale ressaltar que este comportamento já é conhecido na literatura como problema de adaptação de domínio. O qual, segundo Indurkhya e Dumerau (2010), ocorre porque modelos de classificação de sentimentos são altamente sensíveis em relação ao domínio em que os dados utilizados no treinamento foram extraídos. Além disso, Indurkhya e Dumerau (2010) afirmam que a razão para esta sensibilidade dos modelos é causada porque opiniões capturadas em textos de domínios diferentes podem possuir palavras e estruturas de linguagem distintas.

V. Considerações Finais

Este trabalho teve como objetivo propor três modelos de análise de sentimento em textos de avaliações de usuários no site Glassdoor, referente às 105 melhores empresas de TI para se trabalhar no Brasil[1]. Primeiramente, apresenta-se na Tabela XII. Estes modelos foram baseados em RNNs (LSTM, BiLSTM e CNN-BiLSTM), devido a especia-lidade das redes neurais recorrentes em capturar e manter contextos a partir de longas sequências de dados, como em textos (JURAFSKY; MARTIN, 2014).

Com base nos experimentos propostos, foi possível atingir o primeiro objetivo estabelecido no artigo ao revelar a baixa acurácia do modelo ANP (0,3350) em realizar predições com o $dataset$ Glassdoor, ficando abaixo do limiar de 50% de acurácia da hipótese estabelecida no início do estudo. Isto evidenciou a existência de um problema típico de modelos de análise de sentimento que são aplicados em contextos diferentes dos quais foram treinados: o problema de adaptação de domínio (INDURKHYA; DUMERAU, 2010).

Além disso, o segundo objetivo alcançado foi de criar um modelo capaz de superar a acurácia do modelo ANP, sendo que no início do estudo sua acurácia era desconhecida, afinal o $dataset$ Glassdoor foi criado no início dos experimentos. Desta forma, todos os modelos criados conseguiram superar o modelo ANP, atingindo marcas superiores a 0,90 de acurácia.

[1]Fontes dos modelos, *web scraper* e *dataset* disponíveis para *download* no *wiki* do seguinte repositório: https://github.com/wesmaffazzolli/sentiment-analysis-glassdoor-lstm-bilstm-cnnbilstm-

Com isso, alcançou-se o terceiro objetivo proposto que consistia em criar pelo menos um modelo com acurácia acima de 0,70. No entanto, todos os modelos propostos foram capazes de superar esta marca, sendo a maior acurácia alcançada pelo modelo híbrido CNN-BiLSTM (0,9896), seguida pelos modelos LSTM (0,9883) e BiLSTM (0,9847).

Como trabalhos futuros, diversas iniciativas podem ser realizadas para buscar melhoria dos resultados já alcançados. A primeira delas seria realizar o *grid search* com uma maior combinação de parâmetros, os quais tiveram que ser reduzidos para se adequar às limitações de tempo de execução dos ambientes do Google Colaboratory. Uma segunda iniciativa poderia utilizar uma variação modelo do word2vec, como o doc2vec, para realização de extração de características do *corpus*, que atingiu bons resultados em um estudo similar proposto por Rhanoui et al. (2019). Assim como outras abordagens de extração de *features* a partir de textos, amplamente conhecidas na literatura, como o TF-IDF, *Bag-of-words* (BOW) (INDURKHYA; DUMERAU, 2010), o GloVe (JURAFSKY; MARTIN, 2014) e entre outros existentes na literatura.

Além disso, podem ser propostos novos modelos para comparação, como um híbrido de CNN e LSTM (CNN-LSTM), descrito por Rhanoui et al. (2019). Especialmente, o modelo CNN-LSTM pode obter bons resultados neste contexto, já que a LSTM proposta neste artigo superou a BiLSTM. Neste sentido, outro modelo que também pode ser aplicado neste contexto é o da própria CNN (Modelo ANP) proposta pela ETI, o qual pode ser treinado com os dados do próprio $dataset$ Glassdoor.

Outro caminho que pode ser trilhado a partir dos resultados deste estudo é aplicar os modelos em textos reais de canais de comunicação de empresas, por meio da integração dos modelos com sistemas de informação. Com isso, a eficácia dos modelos obtidos neste estudo pode ser colocada à prova diante de diferentes bases textuais rotuladas e, desta forma, pode-se descobrir como que estes modelos reagem em diferentes cenários. Vale ressaltar que a aplicação destes modelos não necessariamente se restringe ao domínio textual referente ao Glassdoor ou outros portais de recrutamento e seleção. Uma vez que, opiniões e sentimentos de colaboradores também podem ser encontrados em $chats$ internos das organizações, pesquisas de clima organizacional, entrevistas de desligamento, e entre outras fontes de textos existentes nas organizações. A partir destas aplicações, também espera-se que os modelos possam apoiar empresas no acompanhamento constante do sentimento nos textos sem a necessidade de realizar tarefas manuais e repetitivas. Uma aplicação prática disso pode ser a integração do modelo em um *dashboard* gerencial, o qual pode estar constantemente monitorando o sentimento das pessoas conforme os textos que publicam e, com isso, e descobrir quais assuntos e tópicos sensibilizam as pessoas, gerando novas correlações antes não conhecidas ou validando hipóteses, por exemplo.

De forma geral, o contexto de análise de sentimentos é vasto e as sugestões apresentadas são alguns dos inúmeros caminhos a serem explorados com base nos resultados alcançados.

REFERÊNCIAS

[1] ALPAYDIN, Ethem. Introduction to Machine Learning. 3. ed. Cambridge: The Mit Press, 2014.

[2] ARMSTRONG, Michael; TAYLOR, Stephen. Armstrong's handbook of human resource management practice. 13. ed. [S.I.]: Kogan Page Limited, 2014. 842 p. Disponível em: https://dl.icdst.org/pdfs/files/8483f557c9bb0435e935b4e9554f5a55.pdf. Acesso em: 29 out. 2021.

[3] AMA, American Marketing Association. Definitions of Marketing. 2017. Disponível em: https://www.ama.org/the-definition-of-marketing-what-is-marketing/. Acesso em: 28 ago. 2021.

[4] AMBLER, Tim; BARROW, Simon. The employer brand. Journal Of Brand Management, [S.L.], v. 4, n. 3, p. 185-206, dez. 1996. Springer Science and Business Media LLC. http://dx.doi.org/10.1057/bm.1996.42. Disponível em: https://www.researchgate.net/publication/263326597_The_employer_brand. Acesso em: 28 ago. 2021.

[5] ARANHA, Christian Nunes. Uma Abordagem de PréProcessamento Automático para Mineração de Textos em Português: sob o enfoque da inteligência computacional. 2007. 144 f. Tese (Doutorado) - Curso de Pós-Graduação em Engenharia Elétrica, Departamento de Engenharia Elétrica do Centro Técnico Científico, Pontifícia Universidade Católica do Rio de Janeiro PucRio, Rio de Janeiro, 2007. Disponível em: https://www.maxwell.vrac.puc-rio.br/colecao.php?strSecao=resultado&nrSeq=10081@1. Acesso em: 29 jun. 2021.

[6] ARORA, Sanjeev; KHODAK, Mikhail; SAUNSHI, Nikunj. A COMPRESSED SENSING VIEW OF UNSUPERVISED TEXT EMBEDDINGS, BAG-OF-n-GRAMS, AND LSTMS. 2018. Disponível em: https://openreview.net/forum?id=B1e5ef-C-. Acesso em: 02 out. 2021.

[7] BACKHAUS, Kristin; TIKOO, Surinder. Conceptualizing and researching employer branding. Career Development International, [S.L.], v. 9, n. 5, p. 501-517, ago. 2004. Emerald. http://dx.doi.org/10.1108/13620430410550754. Disponível em: https://www.emerald.com/insight/content/doi/10.1108/13620430410550754/full/html. Acesso em: 28 ago. 2021.

[8] BENGIO, Yoshua; COURVILLE, Aaron; VINCENT, Pascal. Representation Learning: a review and new perspectives. Ieee Transactions On Pattern Analysis And Machine Intelligence. [S.L.], p. 1798-1828. mar. 2013. Disponível em: https://ieeexplore.ieee.org/document/6472238. Acesso em: 24 out. 2021.

[9] BOJANOWSKI, Piotr et al. Enriching Word Vectors with Subword Information. 2017. Disponível em: https://arxiv.org/pdf/1607.04606.pdf. Acesso em: 31 out. 2021.

[10] BRASSCOM. Relatório Setorial 2020: macrossetor de tic. 4. ed. São Paulo: [S.L.], 2021. 16 p. BRI2-2021-005 - Relatório Setorial 2020. Disponível em: https://brasscom.org.br/relatorio-setorial-2020-macrossetor-de-tic/. Acesso em: 30 jun. 2021.

[11] CASTRO, Leandro Nunes de; FERRARI, Daniel Gomes. Introdução à Mineração de Dados: conceitos básicos, algoritmos e aplicações. São Paulo: Editora Saraiva, 2016. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788547201005/. Acesso em: 19 jul. 2021.

[12] CHAPMAN, Pete et al. CRISP-DM 1.0: step-by-step data mining guide. A: Spss Inc, 2000.

[13] CONNLEY, Courtney. 5 tips for finding and landing a new job in 2021, according to Glassdoor's CEO. 2021. CNBC. Disponível em: https://www.cnbc.com/2021/01/07/5-tips-for-landing-a-new-job-in-2021-according-to-glassdoors-ceo.html. Acesso em: 28 jul. 2021.

[14] DELOITTE. TI tem papel fundamental durante pandemia e na retomada dos negócios. 2020. Disponível em: https://valor.globo.com/patrocinado/deloitte/impacting-the-future/noticia/2020/04/16/ti-tem-papel-fundamental-durante-pandemia-e-na-retomada-dos-negocios.ghtml. Acesso em: 30 jun. 2020.

[15] G1. Abertura de vagas em tecnologia cresce mais de 600% em São Paulo em 2020: veja cargos em alta. 2021. Disponível em: https://g1.globo.com/economia/concursos-e-emprego/noticia/2021/01/26/abertura-de-vagas-em-tecnologia-cresce-mais-de-600percent-em-sao-paulo-em-2020-veja-cargos-em-alta.ghtml. Acesso em: 01 jul. 2021.

[16] GÉRON, Aurélien. Hands-On Machine Learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. California: O'Reilly, 2017.

[17] GLASSDOOR. Best Practices for Employer Branding on Glassdoor Q&A: Part 1. 2014. Disponível em: https://www.glassdoor.com/employers/blog/best-practices-employer-branding-glassdoor-qa-part-1/. Acesso em: 28 jul. 2021.

[18] GLOBO, O. 'E agora, Brasil?': País vive 'evasão silenciosa' de profissionais de tecnologia. 2020. Disponível em: https://oglobo.globo.com/economia/e-agora-brasil-pais-vive-evasao-silenciosa-de-profissionais-de-tecnologia-1-24743726. Acesso em: 27 jun. 2021.

[19] GOASDUFF, Laurence. COVID-19 Accelerates Digital Strategy Initiatives. 2020. Gartner. Disponível em: https://www.gartner.com/smarterwithgartner/covid-19-accelerates-digital-strategy-initiatives/. Acesso em: 30 jun. 2021.

[20] GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel. Data mining: um guia prático. 4. ed. Rio de Janeiro: Elsevier, 2005.

[21] GONÇALVES, Eduardo Corrêa. Mineração de Texto: conceitos e aplicações práticas. Sql Magazine, [S. L], v. 105, p. 31-44, 2012. Disponível em: https://www.researchgate.net/publication/317912973_Mineracao_de_texto_-_Conceitos_e_aplicacoes_praticas. Acesso em: 29 jun. 2021.

[22] GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep Learning. [S.l]: The Mit Press, 2016. 800 p. (Adaptive Computation and Machine Learning series). Disponível em: https://www.deeplearningbook.org/. Acesso em: 19 out. 2021.

[23] GPTW, Editor. Melhores práticas para atração e retenção de talentos para sua empresa. 2020. Disponível em: https://gptw.com.br/conteudo/artigos/atracao-e-retencao-de-talentos/. Acesso em: 13 jul. 2021.

[24] GPTW. Melhores Grandes Empresas para se Trabalhar. 2020. Disponível em: https://gptw.com.br/ranking/melhores-empresas/?ano=2020&tipo=Setorial&ranking=Tecnologia&corte=1000+funcion%C3%A1rios+e+acima. Acesso em: 15 ago. 2021.

[25] GPTW. Melhores Médias Empresas para se Trabalhar. 2020. Disponível em: https://gptw.com.br/ranking/melhores-empresas/?ano=2020&tipo=Setorial &ranking=Tecnologia&corte=100+%C3%A0+999+funcion%C3%A1rios. Acesso em: 15 ago. 2021.

[26] GRAVES, Alex; FERNANDEZ, Santiago; SCHMIDHUBER, Jürgen. Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. 2005. Disponível em: https://www.researchgate.net/publication/221080352_Bidirectional_LSTM_Networks_for_Improved_Phoneme_Classification_and_Recognition. Acesso em: 24 out. 2021.

[27] INDURKHYA, Nitin; DUMERAU, Fred J. Handbook of Natural Language Processing. 2. ed. Boca Raton: Chapman & Hall/Crc, 2010. (Machine Learning & Pattern Recognition Series).

[28] JURAFSKY, Daniel; MARTIN, James H. Speech and Language Processing: an introduction to natural language processing, computational linguistics, and speech recognition. 2. ed. [S. L.]: Pearson, 2014. 1032 p.

[29] KALINSKA-KULA, Magdalena; STANIEC, Iwona. Employer Branding and Organizational Attractiveness: current employees perspective. European Research Studies Journal, [S.L.], v. , n. 1, p. 583-603, 1 fev. 2021. ISMA SYC INT. http://dx.doi.org/10.35808/ersj/1982. Disponível em: https://www.researchgate.net/publication/349385036_Employer_Branding_and_Organizational _Attractiveness_Current_Employees'_Perspective. Acesso em: 28 ago. 2021.

[30] KELLER, Kevin Lane. Strategic Brand Management: building, measuring, and managing brand equity. 4. ed. Essex: Pearson Education, 2013.

[31] KINGMA, Diederik P.; BA, Jimmy Lei. ADAM: method for stochastic optimization. METHOD FOR STOCHASTIC OPTIMIZATION. 2015. ICLR 2015. Disponível em: https://arxiv.org/abs/1412.6980. Acesso em: 22 out. 2021.

[32] KIM, Hannah; JEONG, Young-Seob. Sentiment Classification Using Convolutional Neural Networks. Applied Sciences. [S.I], p. 1-1. jun. 2019. Disponível em: https://www.mdpi.com/2076-3417/9/11/2347#cite. Acesso em: 26 out. 2021.

[33] KWARTLER, Ted. Text Mining in Practice with R. Hoboken: Wiley, 2017.

[34] LI, Lin et al. Text Classification Based on Word2vec and Convolutional Neural Network. In: ICONIP 2018, 25., 2018, Cham. Neural Information Processing. Cham: Springer International Publishing, 2018. p. 450-460. Disponível em: https://link.springer.com/chapter/10.1007/978-3-030-04221-9_40. Acesso em: 02 out. 2021.

[35] LIU, Gang; GUO, Jiabao. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing, [S.L.], v. 337, p. 325-338, abr. 2019. Elsevier BV. http://dx.doi.org/10.1016/j.neucom.2019.01.078. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0925231219301067. Acesso em: 26 out. 2021.

[36] LIU, Pengfei et al. Recurrent Neural Network for Text Classification with Multi-Task Learning. In: TWENTY-FIFTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE (IJCAI-16), 25., 2016, New York. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. Palo Alto: Aaai Press / International Joint Conferences On Artificial Intelligence, 2016. p. 2873-2879. Disponível em: https://arxiv.org/abs/1605.05101. Acesso em: 25 out. 2021.

[37] MARTINEZ-PLUMED, Fernando et al. CRISP-DM Twenty Years Later: from data mining processes to data science trajectories. Ieee Transactions On Knowledge And Data Engineering, [S.L.], v. 33, n. 8, p. 3048-3061, 1 ago. 2021. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tkde.2019.2962680.

[38] MARTINS, Júlio Serafim et al. Processamentos de Linguagem Natural. Porto Alegre: Sagah, 2020. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9786556900575/. Acesso em: 08 jul. 2021.

[39] MIKOLOV, Tomas et al. Efficient Estimation of Word Representations in Vector Space. 2013. Disponível em: https://arxiv.org/abs/1301.3781. Acesso em: 25 out. 2021.

[40] MINAEE, Shervin et al. Deep Learning Based Text Classification: a comprehensive review. A Comprehensive Review. 2021. ArXiv:2004.03705v3. Disponível em: https://arxiv.org/abs/2004.03705. Acesso em: 23 out. 2021.

[41] MINER, Gary et al. Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications. Waltham: Elsevier, 2012. 1053 p.

[42] MORAIS, Edison Andrade Martins. Mineração de Textos. Goiânia: [S.I], 2007. Disponível em: https://ww2.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_005-07.pdf. Acesso em: 29 jun. 2021.

[43] OMS. Coronavirus disease (COVID-19) advice for the public. 2021. Organização Mundial da Saúde. Disponível em: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public. Acesso em: 14 jul. 2021.

[44] RHANOUI, Maryem et al. A CNN-BiLSTM Model for Document-Level Sentiment Analysis. 2019. Disponível em: https://www.mdpi.com/2504-4990/1/3/48. Acesso em: 02 out. 2021.

[45] RAO, Adithya; SPASOJEVIC, Nemanja. Actionable and Political Text Classification using Word Embeddings and LSTM. 2016. Disponível em: https://arxiv.org/abs/1607.02501v2. Acesso em: 02 out. 2021.

[46] ŘEHůřEK, Radim. Word2Vec Model. Disponível em: https://radimrehurek.com/gensim_3.8.3/auto_examples/tutorials/run_word2vec.html. Acesso em: 25 out. 2021.

[47] ŘEHůřEK, Radim. Models.word2vec: word2vec embeddings. Word2vec embeddings. 2021. Disponível em: https://radimrehurek.com/gensim_3.8.3/models/word2vec.html. Acesso em: 25 out. 2021.

[48] ROSA, Bruno; RIBAS, Raphaela. Na falta de profissionais de TI, empresas treinam os funcionários que já têm. 2021. Disponível em: https://oglobo.globo.com/economia/tecnologia/na-falta-de-profissionais-de-ti-empresas-treinam-os-funcionarios-que-ja-tem-25058974. Acesso em: 27 jun. 2021.

[49] RUSSELL, Stuart; NORVIG, Peter. Artificial Intelligence: a modern approach. 3. ed. New Jersey: Pearson Prentice Hall, 2010. (Prentice Hall Series In Artificial Intelligence).

[50] SACHAN, Devendra Singh; ZAHEER, Manzil; SALAKHUTDINOV, Ruslan. Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function. 2020. Disponível em: https://arxiv.org/abs/2009.04007. Acesso em: 02 out. 2021.

[51] SEBRAE. Análise da crise e impactos para os pequenos negócios. Espírito Santo: Sebrae, 2020. 136 p. Disponível em: https://www.sebrae.com.br/sites/PortalSebrae/ufs/es/sebraeaz/analise-da-crise-e-impactos-para-os-pequenos-negocios,

7d521afb0a273710VgnVCM1000004c00210aRCRD. Acesso em: 27 jun. 2021.

[52] SENA, Victor; GRANATO, Luísa. 260.000 vagas sem dono: um raio-x das vagas mais quentes agora (e no futuro). um raio-x das vagas mais quentes agora (e no futuro). 2021. Disponível em: https://exame.com/carreira/260-000-vagas-de-trabalho-sem-dono-conheca-o-setor-que-ganhou-forca-com-a-pandemia/. Acesso em: 26 jun. 2021.

[53] SHANDWICK, Weber; RESEARCH, Krc. THE EMPLOYER BRAND CREDIBILITY GAP: bridging the divide. BRIDGING THE DIVIDE. 2017. Disponível em: https://www.webershandwick.com/wp-content/uploads/2018/04/EmployerBrandCredibilityGap.pdf. Acesso em: 29 jul. 2021.

[54] SHEARER, Colin. The CRISP-DM Model: the new blueprint for data mining. The Journal Of Data Warehousing. Seattle, p. 13-22. 2000.

[55] SCHUSTER, Mike; PALIWAL, Kuldip K.. Bidirectional Recurrent Neural Network. Ieee Transactions On Signal Processing. [S.L.], p. 2673-2681. nov. 1997. Disponível em: https://www.researchgate.net/publication/3316656_Bidirectional_recurrent_neural_networks/citations. Acesso em: 24 out. 2021.

[56] SIAMI-NAMINI, Sima; TAVAKOLI, Neda; NAMIN, Akbar Siami. The Performance of LSTM and BiLSTM in Forecasting Time Series. In: 2019 IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA), Não use números Romanos ou letras, use somente números Arábicos]., 2019, Los Angeles. 2019 IEEE International Conference on Big Data (Big Data). [S.L.]: Ieee, 2019. p. 3285-3292. Disponível em: https://scholars.ttu.edu/en/publications/the-performance-of-lstm-and-bilstm-in-forecasting-time-series-2. Acesso em: 24 out. 2021.

[57] TENSORFLOW. Tf.keras.preprocessing.text.Tokenizer. 2021. Disponível em: https://www.tensorflow.org/api_docs/python/tf/keras/ preprocessing/text/Tokenizer. Acesso em: 28 out. 2021.

[58] UNIT, The Economist Intelligence. The transformation imperative: digital drivers in the covid-19 pandemic. digital drivers in the covid-19 pandemic. 2021. Pesquisa patrocinada pela Microsoft. Disponível em: https://transformationimperative.economist.com/executive-summary/. Acesso em: 21 jul. 2021.

[59] ZHANG, Yanbo. Research on Text Classification Method Based on LSTM Neural Network Model. In: 2021 IEEE ASIA-PACIFIC CONFERENCE ON IMAGE PROCESSING, ELECTRONICS AND COMPUTERS (IPEC), 2021, Dalian. 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC). [S.L.]: Ieee, 2021. p. 1019-1022. Disponível em: https://ieeexplore.ieee.org/abstract/document/9421225. Acesso em: 26 out. 2021.

[60] ZHOU, Chunting; SUN, Chonglin; LIU, Zhiyuan; LAU, Francis C.M.. A C-LSTM Neural Network for Text Classification. 2015. Disponível em: https://arxiv.org/abs/1511.08630. Acesso em: 26 out. 2021.

[61] ZHOU, Peng et al. Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. 2016. Disponível em: https://arxiv.org/abs/1611.06639. Acesso em: 02 out. 2021.