ANÁLISE DE SENTIMENTO DE REDES SOCIAIS POR CLASSIFICADORES MULTIMODAIS DE APRENDIZADO DE MÁQUINA

Breno Vieira Arosa

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Luiz Pereira Calôba
Natanael Nunes de Moura Junior

Rio de Janeiro
Setembro de 2019

ANÁLISE DE SENTIMENTO DE REDES SOCIAIS POR CLASSIFICADORES MULTIMODAIS DE APRENDIZADO DE MÁQUINA

Breno Vieira Arosa

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Luiz Pereira Calôba, Dr.Ing.

Natanael Nunes de Moura Junior, D.Sc.

Prof. Nome do Terceiro Examinador Sobrenome, D.Sc.

Prof. Nome do Quarto Examinador Sobrenome, Ph.D.

Prof. Nome do Quinto Examinador Sobrenome, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2019

Vieira Arosa, Breno

Análise de Sentimento de Redes Sociais por Classificadores Multimodais de Aprendizado de Máquina/Breno Vieira Arosa. – Rio de Janeiro: UFRJ/COPPE, 2019.

X, 46 p.: il.; 29, 7cm.

Orientadores: Luiz Pereira Calôba

Natanael Nunes de Moura Junior

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2019.

Referências Bibliográficas: p. 38 – 46.

1. Análise de sentimento. 2. Processamento de linguagem natural. 3. Redes complexas. 4. Aprendizado de máquina. I. Pereira Calôba, Luiz *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

A alguém cujo valor é digno
desta dedicatória.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ANÁLISE DE SENTIMENTO DE REDES SOCIAIS POR CLASSIFICADORES MULTIMODAIS DE APRENDIZADO DE MÁQUINA

Breno Vieira Arosa

Setembro/2019

Orientadores: Luiz Pereira Calôba
Natanael Nunes de Moura Junior

Programa: Engenharia Elétrica

Nos últimos anos, as redes sociais se tornaram um dos principais meios de comunicação e com isso, houve um aumento da influência que exercem sobre os usuários. Por esse motivo, as mensagens que trafegam por elas passam a ter importância para as mais diversas finalidades como, por exemplo, a avaliação de produtos e eventos. Dentre as possíveis análises, a mineração de opinião é uma das operações com mais aplicações diretas. Nesse sentido, ferramentas de processamento de linguagem natural e de redes complexas são capazes de auxiliar a geração destas análises. Entretanto, o desempenho dessas técnicas, em geral, depende da existência e do volume de bases de treinamento anotadas manualmente, dificultando assim a utilização das mesmas. O presente trabalho aplica métodos de geração de bases de treinamento automatizadas para contornar esse obstaculo e gerar classificadores de análise de sentimento. São avaliados diferentes modelos de classificação textual, tanto lineares, como Naïve Bayes e SVM, quanto por *Deep Learning*, como redes convolucionais e redes recorrentes. Também são analisadas técnicas de redes complexas para caracterização dos usuários das redes, abordando assim diferentes aspectos das informações fornecidas por estas mídias.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

SENTIMENT ANALYSIS OF SOCIAL NETWORKS BY MULTIMODAL MACHINE LEARNING CLASSIFIERS

Breno Vieira Arosa

September/2019

Advisors: Luiz Pereira Calôba
Natanael Nunes de Moura Junior

Department: Electrical Engineering

In this work, we present ...

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Nas últimas duas décadas, as redes sociais se tornaram um dos principais meios de comunicação. Esse crescimento, em parte, se justifica pela massificação do acesso a internet incluindo dispositivos móveis como *smartphones* e *tablets*. Também alavancado pelos avanços computacionais e pelo desenvolvimento acelerado de novas técnicas e algoritmos, o aprendizado de máquina, em especial o processamento de linguagem natural, tem essas redes como importante objeto de estudo.

Desde a chamada Revolução Digital, observamos um progressivo barateamento e facilitação do uso de dispositivos eletrônicos. À medida que essas tecnologias passaram a ser acessíveis, não apenas para as corporações, mas também para os indivíduos, houve um crescente processo de digitalização de diversos aspectos de nossas vidas. Com a comunicação não foi diferente. O email, por exemplo, substitui desde os anos 70 operações que até então eram apenas possíveis de forma analógica, como pelo uso de cartas. Nesse contexto, as redes sociais, ou mídias sociais, abordam aspectos diferentes da comunicação, mais dinâmica e informal.

Apesar de já existirem desde os anos 90, é com a virada do milênio que as primeiras grandes mídias sociais online aparecem, como *LinkedIn*, *MySpace* e *Orkut*. Desde então há um aumento anual da quantidade de seus usuários. Atualmente, estima-se que 3,5 bilhões de pessoas, ou 45% da população mundial utilize pelo menos uma rede social. Este número torna-se ainda mais interessante quando considerado que 4,4 bilhões de pessoas têm acesso à internet. Portanto, quase 80% dos internautas estão em alguma das mídias sociais. No Brasil esses números se acentuam ainda mais; 70% da população tem acesso à internet e 66% utiliza as redes sociais [2].

Além da alta penetração na sociedade, devido à disponibilidade proporcionada pelos dispositivos móveis, os usuários consomem boa parte de seu tempo nessas redes. No mundo, gasta-se em média 2 horas e 16 minutos por dia. Novamente esse número é ainda superior no Brasil, onde a média é de 3 horas e 34 minutos, sendo o segundo país no mundo a usar por mais tempo as redes, ficando apenas atrás das Filipinas.

Essa forte presença fez com que as mídias sociais não impactassem apenas as comunicações. Hoje em dia esses meios também são comumente utilizados para busca de relacionamentos, compartilhamento de notícias, divulgação de serviços, atendimento ao público, entre outros. As informações que trafegam nas redes exercem grande influência na formação de opinião das pessoas, seja ela em relação a um produto, a um evento ou até mesmo temas políticos, como pôde-se observar nas eleições pelo mundo nos últimos anos.

Portanto, a análise dessas informações, presentes nas redes, é importante para as mais diversos aplicações. Contudo, essa grande quantidade de usuários também se reflete no número de dados provindos das mídias sociais. Dentre as estatísticas de uso do ano de 2018 fornecidas pelas próprias redes sociais, tem-se que, diariamente, 300 milhões de fotos são publicadas no *Facebook*, 5 bilhões de vídeos são vistos no *YouTube*, 43 bilhões de mensagens são enviadas no *WhatsApp* e 100 milhões de usuários interagem pelo *Twitter*.

O massivo volume de dados inviabiliza que essas análises sejam feitas manualmente, tornando-se necessário o desenvolvimento de ferramentas capazes de automatizar esse processo. Entram aí as técnicas desenvolvidas pelo campo do Processamento de Linguagem Natural, do inglês *Natural Language Processing* (NLP). Foi a partir do anos 50 que esse termo passou a aparecer como um ramo da Inteligência Artificial. Devido à sua complexidade, a linguagem natural acabou inclusive se tornando um critério de inteligência, como proposto no teste de Turing [3].

O NLP é uma ampla área de pesquisa, abrangendo diferentes estágios da língua, desde os níveis de abstração mais baixos, como o estudo da fonologia e da sintaxe, quanto os níveis de abstração superior, que lidam com a semântica de determinado conteúdo. Neste ramo, busca-se desenvolver métodos capazes de auxiliar e/ou automatizar tarefas como: o reconhecimento de fala, a análise sintática de frases, a extração de entidades, a segmentação por tópicos, entre outros.

Esse conjunto de ferramentas são essenciais para tratar o volume de textos gerado pelas redes sociais. Com o avanço de técnicas de *Deep Learning*, nos últimos anos pôde-se observar um avanço significativo de desempenho, que permitiu até que tarefas de grande dificuldade, como a automatização de traduções e de aplicações de atendimento ao cliente por conversação, sejam possíveis.

Entretanto, apesar do grande potencial dessas técnicas, as mídias sociais apresentam características que diferenciam seu conteúdo dos tipos textuais que tradicionalmente se analisam em NLP, como artigos e textos jornalísticos. Isso dificulta o processamento dessa informação. Em geral, as redes sociais se apresentam como meio de conversação ágil, logo, as mensagem que circulam por elas costumam ser extremamente curtas, com amplo uso de abreviações. Por seu caráter informal, observa-se uma alta taxa de erros gramaticais e uma grande utilização de *emoti-*

cons [1]. Somado a isso, o fato de serem meios de comunicação globais também ressalta a presença de estrangeirismos [2]. Ademais a dinamicidade das redes sociais faz com que a evolução de sentido das palavras seja acelerada. Esses elementos trazem a necessidade de adaptação ou desenvolvimento de novas técnicas para se reproduzir o sucesso obtido pelas técnicas de processamento de texto em documentos com escrita mais formal e estruturada.

Porém, o principal fator que distingue as informações de redes de outros meios é a forte interligação entre diferentes tipos de mídia, como textos, imagens, áudios, fotos e vídeos. Além de metadados importantes, como localização, data e horário, uma propriedade importante das mídias sociais são os atributos referentes às redes de usuário. Exemplos desses dados são o número de amizades de um usuários da rede, e o número de re-compartilhamentos de uma mensagem. Logo, apesar da capacidade das ferramentas de NLP, existe um conjunto de informações que essas técnicas desconsideram, abrindo espaço para que as abordagens sejam multimodais, ou seja, que explorem diversas destas propriedades.

De certa maneira, as interações entre usuários são o cerne das mídias sociais. Logo, técnicas que se dispõem a analisar esse tipo de informação também são de grande relevância. Nesse quesito, o campo das Redes Complexas, ou Ciência de Redes, é responsável por estudar os algoritmos e comportamentos observados em grafos que representam sistemas reais, como no caso das redes sociais. Assim como o aprendizado de máquina, essa esfera do conhecimento também apresenta um grande crescimento nos últimos anos, fornecendo um novo leque de tecnologias, de forma a descobrir-se aplicabilidades até então inexploradas. Dentre suas aplicações, que possuem importância para o estudos das mídias sociais, podemos ressaltar, por exemplo, a detecção de comunidades, a identificação de principais influenciadores, e a modelagem de propagação de informação.

Finalmente, estes métodos são poderasas ferramentas de análises de redes sociais, principalmente quando aplicados em complemento à informação textual. A complementariedade entre esses métodos permite, por exemplo, distinguir conotações de uma mesma mensagem quando escritas por usuários de comunidades de ideias opostas. Por causa desses casos complexos, nos quais a análise de um elemento único da mensagem não é suficiente para sua classificação é que estudos que se adequam a multimodalidade das redes sociais se fazem necessários.

[1] Sequência de caractéres ou pequena imagem que transmite um estado emotivo

[2] Uso de palavras ou expressões estrangeiras

1.1 Motivação

Dados são considerados um dos bens mais valiosos da atualidade, de forma mesmo a serem chamados de "o novo petróleo". Isso porque, assim como o óleo, os dados são preciosos e precisam ser refinados para terem utilidade. Um dos aspectos dessa transformação pode ser observado na mudança cultural de organizações e empresas que passam a tomar decisões baseadas em dados e métricas coletadas.

A busca por informação de qualidade sobre um serviço ou produto sempre foi importante para consumidores. Quando não havia as tecnologias usadas atualmente essas pesquisas eram feitas majoritariamente no boca-a-boca ou a partir de revistas especializadas. Com a criação da internet e das redes sociais estas passaram também a exercer essa função, com o beneficio de se encontrar opiniões de forma espontânea e em grande quantidade. As mídias sociais se tornaram um dos principais meios de compartilhamento dessa informação. As empresas, por sua vez, têm a oportunidade de utilizar as opiniões que trafegam nas redes para identificar falhas em suas mercadorias, melhoras sua segmentação, planejar novos produtos, entre outras atividades. Com o fácil acesso a coleta desses dados, as ferramentas capazes de extrair o sentimento dessas mensagens tornam-se fundamentais para viabilizar esse procedimento na escala em que ocorrem.

Apesar das dificuldades inerentes a classificação de mensagens de redes sociais técnicas de aprendizado de máquina, sobretudo *Deep Learning*, e de Redes Complexas apresentam êxito em várias tarefas realizadas sobre elas. Entretanto, o sucesso desses modelos, em geral, depende da quantidade de dados anotados disponíveis para treinamento. Como o processo de anotação é manual esse passa a ser o gargalo da construção de classificadores de sentimento.

Esse empecilho se torna ainda mais notável quando consideramos aplicações que requerem a análise de múltiplas línguas ou que tenham foco em um tema específico, necessitando criação de bases de dados próprias para cada caso de uso. Esses fatos motivam a elaboração de métodos que sejam independentes de bases de treinamento.

1.2 Objetivo

Esse projeto visa desenvolver um método capaz de formar classificadores de análise de sentimento sem a necessidade de bases de dados de treinamento. Essas análises serão feitas sobre dados de mídia sociais e serão explorados atributos tanto textuais quanto das redes de usuários. A principal meta desse trabalho é viabilizar o emprego de modelos complexos, que se beneficiam do grande volume de dados, sem o custo proveniente da anotação de bases de treinamento.

Para análise das mensagens será avaliado o impacto da utilização classificadores

de *Deep Learning* em comparação a métodos lineares tradicionalmente aplicados em NLP. Diferentes estratégias de representação de palavras e arquiteturas de classificadores redes de aprendizado profundo serão testados de acordo com o estado da arte em processamento de texto. O processo será feito de maneira semi-supervisionada com supervisão distante para anotação automática dos dados de treinamento.

Técnicas de Redes Complexas serão aplicadas para caracterização de autores das mensagens. Serão comparados modelos de representação de nós em grafos, támbem baseados em aprendizado de máquina e de treinamento não supervisionado, cujo custo computacional permitam operar em volumes de dados compatíveis com os de redes sociais. Neste caso, além de avaliar os modelos entre si, será analisado a informação adicional provinda do usuários quando aplicada em conjunto com o classificador textual decorre em algum benefício de performance do sistema.

Concluindo, há um amplo conjunto de estudo aplicando de processamento de linguagem natural em redes sociais. Apesar de menor, a ciência de redes também têm um grande reportório de pesquisa sobre esse meio de comunicação. Este trabalho visa preencher a lacuna de sistemas que não necessitam de investimento em anotação de dados e que abrangem a multimodalidade da informação característica desse tipo de dado.

1.3 Organização do Texto

Esse documento é organizado da seguinte maneira:

- O Capítulo 2 apresenta o problema da análise de sentimento aplicada em mídias sociais e seus desafios. Esse Capítulo contém uma breve revisão bibliográfica de classificadores de análise de sentimento e de técnicas de redes aplicadas a essa tarefa.

- No Capítulo 3 as técnicas de processamento de linguagem natural são apresentadas. São caracterizadas as diferentes formas de representação numérica das mensagens, e sobre quais premissas se baseiam. Também são descritos os classificadores, tanto os métodos lineares tradicionalmente aplicados a textos quanto os de *Deep Learning*.

- As ferramentas de ciência de redes são apresentadas no Capítulo 4. Nesse Capítulo são mostrados as diferentes técnicas de representação de vértices de uma rede, que serão aplicadas na modelagem de usuários de mídias sociais.

- O Capítulo 5 descreve o método proposto para desenvolvimento dos classificadores. São apresentados as etapas de formação de bases de dados, de anotação automática da mesma e de classificação.

- Os resultados dos obtidos pelos experimentos propostos são mostrados no Capítulo 6.
- Por fim, o Capítulo 7 avalia os resultados obtidos, apresenta as conclusões e enumera possíveis desdobramentos do trabalho realizado.

Capítulo 2

Análise de Sentimento de Redes Sociais

Análise de Sentimento é o campo de estudos que analisa a opinião, sentimento, atitude e emoções de pessoas em relação a entidades. Essas entidades podem ser pessoas, eventos, produtos, tópicos, entre outros. Na literatura também se encontram os seguintes nomes relacionados a esse ramo: *mineração de opinião*, *extração de opinião*, *mineração de sentimento*, *análise de subjetividade*, *análise de emoção* e *extração de críticas* [4]. Por ser aplicado na maioria das vezes a textos escritos, este campo é filiado ao processamento de linguagem natural, sendo uma de suas ramificações mais ativas.

Segundo CAMBRIA [5], a Análise de Sentimento é dividida entre duas principais tarefas, a extração de polaridade e a detecção de emoções. Enquanto a primeira foca em discernir conteúdo positivo de negativo, a segunda é responsável por classificar em emoções como: felicidade, medo, raiva, tristeza, entre outros.

A autora LIU [4], por sua vez, ressalta que há diferentes níveis de granularidade que essas para execução desta tarefa, a escolha do nível a ser utilizado depende da finalidade pela qual se aplicará a classificação e será uma das principais caraterísticas para definir o tipo de técnica empregada. A Análise de Sentimento pode ser realizada a nível de documento, no qual um documento, como uma avaliação de produto, é avaliado como um todo. Nesses casos se assume implicitamente que um documento expressa uma opinião sobre uma única entidade, como o produto em questão no exemplo citado, também fica implícita que o documento expressa um sentimento único sobre a entidade. Por serem textos mais extensos, logo com mais informação, a assertividade de modelos nesses casos em geral é mais alta, portanto, até técnicas mais simples como as baseadas em dicionario como será apresentado na seção 3.3.1 podem apresentar resultados suficientemente bons. TABOADA *et al.* [6] exemplifica a extração de opinião de documentos a partir de técnicas de dicionário aplicadas em diferentes bases de dados de avaliações de produtos. Por sua vez, DAS e CHEN [7]

cacula predição de valor de ações a partir de análise de sentimento das mensagens presentes em um fórum online de investidores.

Visto que as limitações decorrentes de se classificar documentos por inteiro reduzem o escopo de aplicações, pode-se recorrer ao nível seguinte de granularidade, a classificação de sentimento de sentenças. A principal diferença entre essa abordagem e a anterior é a quantidade de informação disponível dado que uma sentença é composta, geralmente, por poucas palavras. Por outro lado, a premissa de sentimento único no conteúdo de uma frase é mais coerente com a realidade comparando-se com a classificação documento como um todo, sendo uma aproximação boa o suficiente para uma nova gama de casos de uso. Entretanto, LIU [4] ressalta que para o caso de sentenças é importante que a classificação de polaridade leve em consideração o sentimento neutro. Isso se torna relevante pois até mesmo dentro de documentos opinativos, como avaliações de produto, há sentenças puramente objetivas, que não expressarão polaridade sobre uma entidade. RILOFF *et al.* [8] apresentam o primeiro trabalho especificamente voltado para classificação de subjetividade de documentos. Devido a quantidade limitada de informação presentes em uma sentença, classificadores baseados em dicionários e em técnicas de aprendizado de máquina por modelos lineares, apresentam indicadores piores na execução da tarefa. São nesses cenário que nos últimos anos os modelos não lineares começaram a sobressair, como apresentam SOCHER *et al.* [9] e SOCHER *et al.* [10] que aplicam diferentes técnicas baseadas em *Deep Learning* para classificação de sentenças.

Por fim, apresenta-se a análise de sentimento de características. Uma entidade pode ser composta de diversos atributos. Ao falar sobre um filme pode se avaliar diferentes aspectos dele, como o roteiro, os atores, os personagens, etc. Uma crítica a esse determinado filme é composta de sentimentos distintos para cada um desses atributos, e o objetivo da análise de sentimento de características é identificar a polaridade de uma mensagem em relação aos atributos presentes. Para realizar essa análise são necessários elementos novos, como o reconhecimento da entidade citada e quais aspectos dela estão sendo avaliados. Também podem ser relevantes a identificação do autor e do momento do documento analisado. NASUKAWA e YI [11] e SNYDER e BARZILAY [12] são exemplos de trabalhos focados em mineração de opinião focada em aspectos.

A Análise de Sentimento aplicada a redes sociais, em especial ao Twitter, se assemelha a classificação de sentimento de sentenças. Entretanto o perfil de mensagem que circulam as mídias sociais apresentam peculiaridades quando comparadas a meios convencionais. Por se tratar de um ambiente informal e de comunicação rápida, é comum encontrar erros gramaticais e abreviações. Similarmente os *emoticons* também tem ampla adesão por serem métodos práticos de exprimir sentimentos em poucos caracteres. Por se tratarem de redes globais, é frequente o emprego de

palavras ou expressões de outras línguas em uma mesma mensagem. Esses fatores são obstáculos aos classificadores de linguagem natural, dificultando a tarefa de extração de polaridade. A Tabela 2.1 mostra exemplos de mensagens extraídas da rede que demonstram esse problema.

Fator	***Tweet***
Ironia	Recomendo chegar para dar aula e descobrir que mudaram seu horário sem avisar.
Ambiguidade	Estou igualmente fascinada e enojada.
Multiplicidade de idiomas	Macarrão de arroz is the new miojo.

Tabela 2.1: Dificuldades encontradas na classificação de sentimento.

O Twitter é uma rede social baseada em interações por mensagens curtas. Suas principais características são a brevidade e instantaneidade das mensagens, também chamadas *tweets*, que são limitadas atualmente em 280 caracteres mas famosas pelo seu limite anterior de 140 caracteres. Criada em 2006, a rede conta com 139 milhões de usuários ativos diários segundo seu relatório trimestral para investidores [13] e está entre as redes sociais com maior número de usuários do mundo.

O Twitter foi responsável pela criação e popularização das *hashtags*, mecanismo que funciona como marcação de palavra-chave ou tópico. Seu funcionamento se dá pela utilização do simbolo da tralha (#) e pela ausência de espaço e pontuação nos casos que são formados por mais que uma palavra. As *hashtags* funcionam como um agregador de *tweets* e o site ainda apresenta uma lista das mais populares no momento. O sucesso dessa funcionalidade fez com que a mesma fosse posteriormente aderida por outras mídias sociais como o Facebook e Instagram se tornando um atributo marcante mensagens de redes sociais.

Além das *hashtags*, as mensagens no Twitter também possibilitam a inserção de mídias como imagens, vídeos e áudios. Outro atributo comum em mídias sociais são as redes formadas pela conexão de usuários. Essas redes podem ser formadas por relações de amizade, seguidores e outras interações entre essas pessoas ou entidades. O re-compartilhamento de mensagem, que se dá quando um usuário divulga em seu perfil uma mensagem criada por outra pessoa é outro componente capaz de gerar grafos de usuários, no Twitter o re-compartilhamento também é chamado de *retweet*.

Como citados anteriormente, textos que circulam as mídias sociais contém desafios adicionais em comparação a documentos tradicionalmente estudados pela área de NLP, como artigos jornalísticos e avaliações de produto. Apesar disso, técnicas de classificação de sentimento puramente textuais obtiveram uma crescente melhora de desempenho, principalmente com a aplicação de técnicas de *Deep Learning*.

Um dos primeiros e mais influentes trabalhos sobre aplicação de aprendizado de

máquina para análise de sentimento foi feito por PANG *et al.* [14], que comparou técnicas de classificação feitas a partir de dicionário com modelos de *Naive Bayes*, Máxima Entropia e Máquinas de Vetor Suporte (SVM). Em seu estudo PANG *et al.* [14] também compara a eficácia de diferentes métodos de representação do texto para o uso de modelos de aprendizado de máquina.

Inspirado nos métodos de PANG *et al.* [14], GO *et al.* [1] utilizou as mesmas técnicas para classificação de sentimento em Twitter. A grande diferença entre os trabalhos de PANG *et al.* [14] e GO *et al.* [1] é que enquanto no primeiro caso o problema de classificação de críticas a filmes a coleta de dados já fornece uma anotação, dado que as críticas eram acompanhadas de um sistema de avaliação (entre 1 e 5 estrelas), o posterior não contou com disponibilidade parecida e, por isso, aplicou um método de anotação automática.

A maneira convencional de abordar bases de dados não anotadas é realizar uma classificação manual dos dados. Por depender de iteração humana torna essa uma das etapas mais custosas do processo de criação de classificadores. Além disso, a alta dinamicidade dos temas e do vocabulário presente nas redes sociais faz com que o prazo em que a base de dados anotada seja relevante seja menor. Considerando também que um maior volume das bases de dados, em geral, permitem os modelos treinados a obter melhores resultados, a soma desses fatores torna ineficiente o processo de anotação manual.

O processo utilizado por GO *et al.* [1] foi criado por READ [15] e denominado supervisão distante. Este método consiste em utilizar alguma característica que tenha alta correlação com a predição desejada e utilizá-la para anotar a base de dados. No caso dos trabalhos citados foram selecionados conjuntos de *emoticons* positivos e negativos que serviram para a anotação ruidosa. Desta forma a restrição para construção de bases de dados passa a ser a coleta de *tweets* e recursos computacionais. Em GO *et al.* [1] a base de treinamento, a qual foi anotada por supervisão distante, conteve 800 mil *tweets*. Para fins de validação do modelo ainda foi necessário elaborar uma base de dados anotada manualmente, sendo esta de apenas 359 *tweets*.

Emoticons Positivo	**Emoticons Negativos**
:)	:(
:-)	:-(
:)	: (
:D	
=)	

Tabela 2.2: Emoticons selecionados por GO *et al.* [1] para supervisão distante de *tweets*.

Tweet	**Classe**
Adoro aquelas amizades que alinham em tudo mesmo em cima da hora :)	Positivo
Fico mais triste ainda porque tenho android :(	Negativo
To com um mal estar tão grande no corpo desde ontem :-(zero forças	Negativo
Muito feliz com minha nova tattoo :D	Positivo

Tabela 2.3: Exemplos de *tweets* anotados por supervisão distante com emoticons selecionados da Tabela 2.2.

Entre as desvantagens dessa prática estão o fato de o conjunto de *emoticons* selecionados para anotação ruidosa precisar ser removidos dos dados de treinamento do modelo para não introduzir viés na classificação. A qualidade da supervisão distante também depende da seleção dos *emoticons*. Ademais, não se pode descartar que é possível uma classe tenha subclasses que não sejam correlacionadas com a característica escolhida para executar a supervisão distante. Por exemplo, dentro do conjunto de *tweets* positivos, a subclasse de *tweets* irônicos positivos, pode não conter *emoticons*, e assim não estar presente no conjunto de treinamento. Apesar dessas limitações, a anotação automática foi fundamental para alavancar a performance dos classificadores de sentimento de redes sociais.

A forma de se transformar o corpus textual em números foi objeto de muitos estudos. Como alternativa ao *bag-of-words*, WANG e MANNING [16] apresentaram um método de utilizar pesos obtidos a partir do treinamento de um modelo de *Naive Bayes* como entrada de um classificador de SVM. Por sua vez, PALTOGLOU e THELWALL [17] estudaram variações de técnicas de Recuperação de Informação com a mesma finalidade. Outro eixo importante foram as representações densas, ou de baixa dimensionalidade. Os principais trabalhos relacionados a essas representações foram feitos por MIKOLOV *et al.* [18], conhecido como *Word2Vec*, PENNINGTON *et al.* [19], denominado *GloVe* e mais recentemente *ELMo* produzido por PETERS *et al.* [20] e *BERT* de DEVLIN *et al.* [21].

A disposição de grandes bases de dados e as representações densas foram a condição *sine qua non* para viabilizar a aplicação de modelos de *Deep Learning* em análise de sentimento. *Deep Learning*, ou Aprendizado Profundo, foi responsável por romper barreiras de performance de aprendizado de máquina em diversas áreas [22] como classificação de imagem [23], reconhecimento de fala [24], detecção de doenças [25]. O processamento de linguagem natural foi um dos campos mais impactados por essa linha de técnicas, seja na realização de traduções [26], na correção de erros gramaticais [27], no reconhecimento de entidades [28], na criação de resumos [29], entre outros. Dentre as aplicações de *Deep Learning* em análise de

sentimento temos KIM [30] com a utilização de redes neurais convolucionais para classificação de sentenças, ZHOU *et al.* [31], por sua vez, demonstraram o uso de redes LSTM, baseadas em redes recorrentes, e, SOCHER *et al.* [10] aplica um modelo recursivo.

Entretanto, a mistura entre diferentes modalidades de comunicação: texto, imagem, video, etc gera uma dificuldade adicional para análise das mensagens. O texto presente em um *tweet* que possuí uma imagem pode fazer interlocução com a mesma. Uma mensagem enviada por um usuário pode ter sentido oposto a mesma mensagem quando comunicada por alguém que pertença a um grupo opositor. Nesses casos a análise do conteúdo textual por si só é incapaz de captar a essência da mensagem. Dá-se assim a necessidade de abordagens multimodais para qualquer tipo de análise de redes sociais.

Como a interação entre usuários é uma das principais componentes das mídias esse se torna um objetivo natural para se estudar em complemento ao texto. Para explorar essa componente é possível aplicar ferramentas desenvolvidas pela área de pesquisa de Redes Complexas, também chamada de Ciência de Redes. Assim como o Aprendizado de Máquina, o ramo de estudo de Redes Complexas apresenta um grande crescimento nos últimos anos dado a variedade sistemas em que suas análises e modelos são aplicados com efetividade [32]. Entre os exemplos de sucesso estão a identificação de doenças [33], predição de propagação de epidemias [34], estudo de robustez de redes de roteadores [35] e identificação de operadores de lavagem de dinheiro [36].

No caso da aplicação dessas técnicas em redes sociais podemos citar RATKIEWICZ *et al.* [37] que usa características da rede formada por usuários do Twitter que interagiram com *tweets* sobre as campanhas eleitorais de 2010 dos Estados Unidos para detectar a propagação orquestrada de conteúdo. VAROL *et al.* [38] utilizou características como densidade da rede, coeficiente de clusterização, entre outros atributos não relacionados aos grafos para detecção de contas de usuário robôs. BACKSTROM e LESKOVEC [39] aplicou passeios aleatórios com pesos definidos por atributos da rede de amizades para sugerir conexão entre dois usuários de uma rede social. Por sua vez, QIU *et al.* [40] desenvolveram uma técnica de representação de nós baseada em redes neurais convolucionais para prever a influência de usuários em redes sociais. Os exemplos citados reforçam que ferramentas de Ciência de Redes conseguem acessar informação codificada pela rede de usuários de mídias sociais e que essa informação é útil para diversas aplicações.

Capítulo 3

Processamento de Linguagem Natural

Neste capitulo serão apresentadas as técnicas de Processamento de Linguagem Natural que compõe um classificador, como o de análise de sentimento. De modo geral, esse processo é composto de 3 etapas, como demostrado no diagrama 3.1. As seções a seguir descrevem cada uma destas fases.

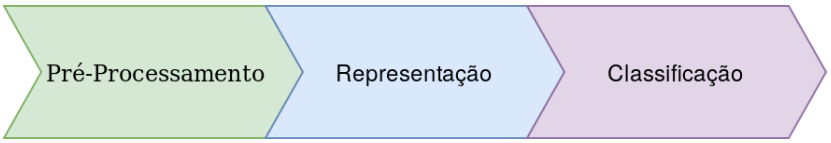


Figura 3.1: Etapas de algoritmos de Processamento de Linguagem Natural.

3.1 Pré-Processamentos

A primeira etapa aplicada para elaboração de modelos de NLP é o pré-processamento. Esta fase consiste na limpeza e preparação dos dados, visando melhorar a performance do classificador seja retirando ruídos dos textos, reduzindo o tamanho do vocabulário ou formatando o texto de maneira a facilitar a modelagem. O volume do vocabulário considerado é costuma ser limitado seja pelos recursos computacionais quanto pelo requisito mínimo de estatística das palavras na base de dados. Portanto, técnicas de pré-processamento que reduzam o tamanho total do vocabulário tem um importante papel na garantia de bom funcionamento dos classificadores.

Como a maior parte dos modelos de NLP trabalham a nível de palavra é necessário separar separar o documento em frases, com algoritmos como o Punkt [41], e posteriormente, em palavras. Esse processo chamado *tokenização* precisa ser ro-

bustos a abreviações, números e características do idioma ao qual será aplicado, como contração de palavras. Se tratando de redes sociais, também é relevante tratar os links, as *hashtags* e as menções a usuários.

Algoritmos de correção ortográfica [42][43] podem ser eficientes para aprimorar a qualidade dos textos, principalmente se tratando de meios de comunicação dinâmicos como as redes sociais.

Técnicas de stemização consistem na extração do o radical das palavras, como o obtido pelo algoritmo de Porter [44], um exemplo é dado com a palavra "montanha"que possuí radical "mont", o mesmo obtido pela palavra "monte". Por outro lado, o processos de lematização tem finalidade parecida, porém transforma a palavra em sua forma base, forma como elas aparecem no dicionario, podendo então diferenciar palavras com o mesmo radical, como "banco"e "bancários". Ambas as técnicas visam tornar as etapas posteriores menos sensíveis a flexões gramaticais, além de colaborar na redução do vocabulário.

Uma das principais etapas do pré-processamento é a remoção das *stopwords*, conjunto de palavras que informação pouco discriminante para uma dada aplicação [45], geralmente são compostas pelas palavras mais comuns da língua, principalmente artigos e preposições. O objetivo da remoção das *stopwords* é diminuir ruídos dos dados textuais, assim simplificando a etapa de modelagem. SAIF *et al.* [46] fizeram um estudo comparando diversos métodos de seleção de *stopwords* e o impacto das mesmas na classificação de sentimento de *tweets*. A identificação de classe gramatical, em inglês *part of speech*, além de ser tipicamente utilizada como entrada de modelos de NLP, também pode ser útil para selecionar *stopwords*.

3.2 Representações

Uma vez que os tratamentos iniciais dos textos são feitos chega-se a etapa de preparar esses dados para serem processados pelo modelo. Para isso, os documentos são transformados de sequências de palavras em vetores ou matrizes. Há diversas técnicas desenvolvidas com essa finalidade, estas podem ser divididas em representações esparsas e representações densas. As representações esparsas são as mais simples de serem aplicadas dado que, em geral, não dependem do treinamento de nenhum modelo. Entretanto, as representações esparsas resultam em vetores ou matrizes de dimensão na ordem de, pelo menos, o tamanho do vocabulário escolhido, como os vocabulários costumam ser muito extensos (centenas de milhares de palavras em alguns casos) o tamanho e a esparsidade da representação obtida podem dificultar o treinamento dos classificadores. Para contornar essa dificuldade foram criados algoritmos de representações densas. Por transformarem os documentos em vetores ou matrizes de baixa dimensão estes algoritmos foram os responsáveis pela

viabilidade de utilização de técnicas de *Deep Learning* aplicadas ao processamento de linguagem natural. Nessa seção descreveremos algumas das principais técnicas de representação de texto.

3.2.1 Codificação One-Hot

A codificação *One-Hot* representa cada palavra de maneira maximamente esparsa. Para tal, é definido um espaço vetorial em que cada palavra do vocabulário utilizado é equivalente a uma dimensão do espaço. Portanto, um documento pode ser transcrito dessa forma em uma sequência de vetores, ou matriz, em que cada palavra é um vetor com valor unitário na dimensão da própria palavra e zero nas outras, como mostra a Figura 3.2.

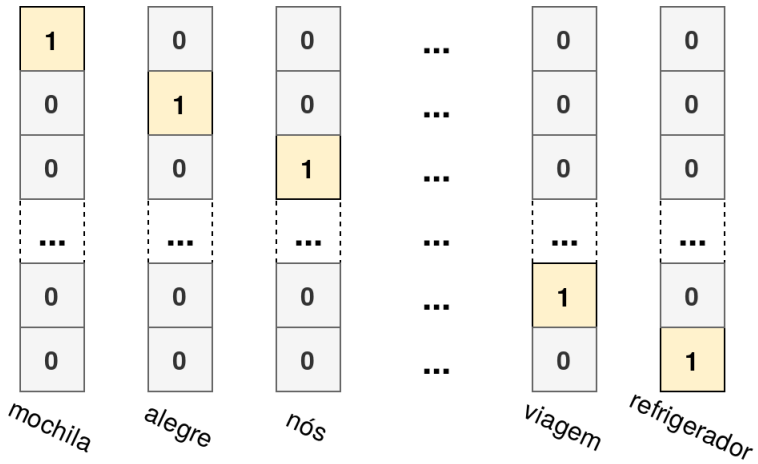


Figura 3.2: Vetores *One-Hot* de palavras de um dicionário. A dimensionalidade do vetor é igual ao tamanho do dicionário.

Frequentemente encontramos nas línguas palavras compostas ou expressões. Essas informações são perdidas na codificação *One-Hot*. Uma forma de se atenuar esse problema são com os chamados *n-gramas*. A ideia do *n-grama* é formar tokens de 2, 3, ou n palavras e utiliza este conjunto de palavras como dimensão do espaço. Entretanto, o aumento no número de palavras por token também gera um aumento significativo do número de dimensões, dificultando o treinamento do classificador.

3.2.2 Bag-of-Words

A codificação *Bag-of-Words* é uma alternativa a representação de mensagens como matrizes compostas de vetores *One-Hot* de suas palavras. Esta é feita pela soma destes mesmos vetores [47]. Portanto, a representação final é dada por um único vetor, de tamanho correspondente ao do vocabulário utilizado. Esta técnica também tem a vantagem de transformar documentos de tamanho variados em vetores de

mesmas dimensões, fator que precisa ser contornado em codificações baseadas em palavras.

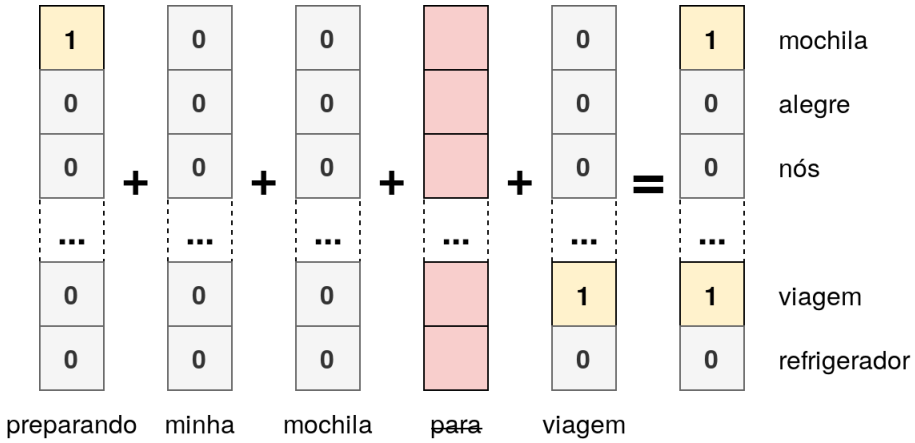


Figura 3.3: Processo de representação por *Bag-of-Words* da frase "Preparando minha mochila para viagem.". A palavra "para" é removida durante o pré-processamento por ser uma *stopword*.

Entretanto, visto que a distribuição de palavras em um corpus, em geral, segue a lei de Zipf [48], ou seja, sua frequência segue uma distribuição em lei de potência. Neste caso mesmo retirando *stopwords*, as palavras mais comuns do vocabulário ainda dominarão os documentos, e este comportamento pode ser prejudicial para o treinamento dos modelos que serão menos expostos a palavras incomuns.

Para atenuar esse problema pode-se aplicar o *term frequence-inverse document frequence* (TF-IDF) [49]. Neste caso, a representação segue a mesma estrutura proposta pela codificação *bag-of-words*, entretanto, cada palavra é ponderada por um multiplicador inversamente proporcional a sua frequência nos documentos.

Bag-of-words e TF-IDF são métodos de representação muito presentes na literatura pela sua simplicidade de implementação e pelos benefícios anteriormente descritos, em especial em conjunto com modelos como a SVM 3.3.2 que apresenta menos dificuldades de ser treinada em dados esparsos. Entretanto, um componente fundamental da linguagem é perdido nesse processo, o contexto em que se insere cada palavra. Pela representação agrupar todos os tokens em um único vetos a ordem das palavras no documento é perdida, o que em muitos casos pode corresponder na inviabilidade de uma classificação precisa do mesmo.

As técnicas de representações densas que serão apresentadas a seguir além de resolverem o obstáculo da dimensionalidade dos dados também viabilizam a classificação por modelos que levem em conta o contexto de cada palavra.

3.2.3 Word2Vec

Word2Vec [18] foi uma das primeiras técnicas de representação densa de palavras amplamente adota pela indústria e academia. Representações densas são aquelas em que cada palavra é transformada em um vetor de números reais de baixa dimensionalidade, tipicamente dezenas ou poucas centenas de dimensões. MIKOLOV *et al.* [18] mostraram que essa representação é capaz de capturar parte dos sentidos semânticos e sintáticos das palavras, aproximando as que são sinônimas ou exerçam a mesma função gramatical, atenuando assim o problema da disparidade de frequência das palavras.

Esta técnica é um modelo constituído de uma rede neural de uma camada escondida. Para o treinamento do mesmo são feitas janelas de um número arbitrário de palavras que servirão como entrada do modelo. Dessas janelas há duas variações: o *continuous bag-of-words* em que o modelo é treinado para prever a palavra central da janela de entrada a partir das outras palavras da janela, ou o *skipgram* que a partir da palavra central da treina-se para prever o seu contexto. A Figura 3.4 ilustra a diferença entre os treinamentos. As entradas e saídas do modelo são representadas pela codificação *one-hot* dos termos. A quantidade de neurônios escolhido para a camada escondida da rede resultará na dimensionalidade da representação. A representação das palavras serão os pesos da camada de entrada do modelo treinado.

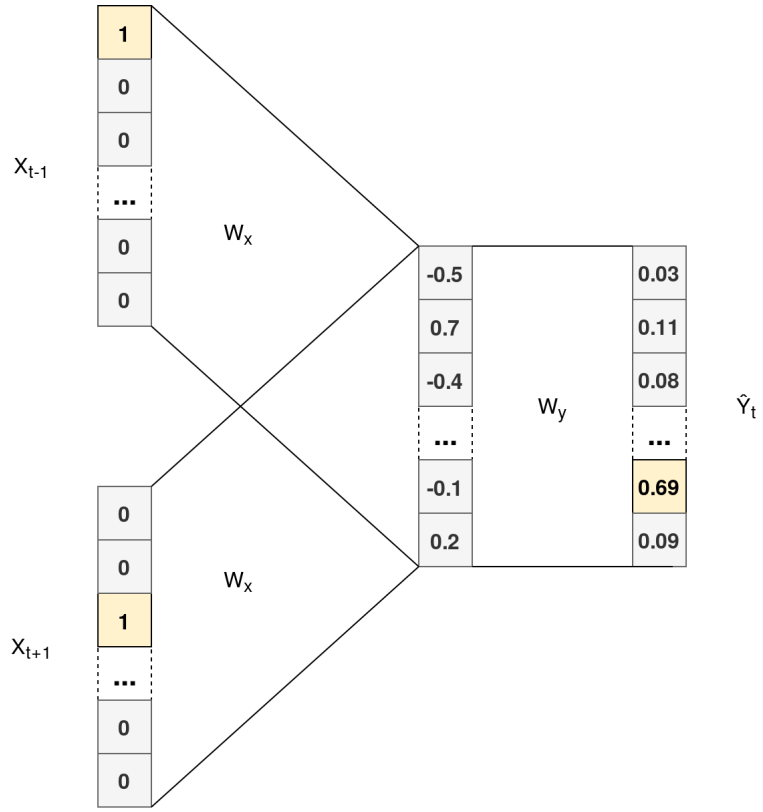
Apesar do *Word2Vec* ser um modelo que precisa ser treinando, esse treinamento é não-supervisionado dado que tanto as entradas quanto saídas do modelo são obtidas diretamente dos documentos. Desta forma, foi possível aplicar o algoritmo a grandes bases de dados mineiradas da internet. Esse alto volume de dados de treinamento foi essencial para que bons resultados fossem alcançados.

Por ser a primeira representação densa de sucesso, o *Word2Vec* foi essencial para a aplicação de classificadores também baseados em redes neurais, como os de *Deep Learning* que obtiveram grande êxito em diversas tarefas de processamento de linguagem natural. Outras técnicas semelhantes e também amplamente adotadas na indústria foram desenvolvidas neste mesmo período de tempo como o GloVe [19] e o FastText [50].

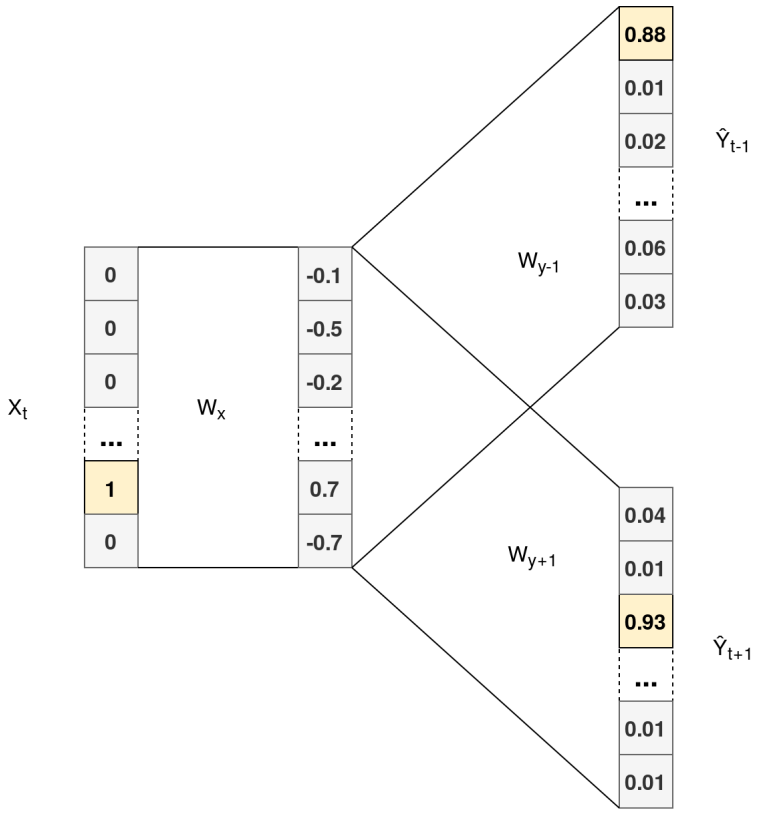
3.2.4 Representações por Redes Neurais Recorrentes

O sucesso do *Word2Vec*, baseado em redes neurais *feed forward* inspirou a experimentação de outros tipos de redes neurais, dentre as quais as redes neurais recorrentes e suas variações obtiveram bons resultados na representação de palavras. Nesta seção serão descritos alguns destes algoritmos.

A estrutura chamada *Enconder-Decoder* desenvolvida por CHO *et al.* [51] se tornou a base das principais representações por redes recorrentes. Esta estrutura



(a) Continuos Bag-of-Word



(b) Skipgram

Figura 3.4: Esquemas de treinamento do Word2Vec. A matriz X_t representa a palavra da janela na posição t e $\hat{Y}_t$ a predição do modelo para a mesma, W, por sua vez, o conjunto de pesos treináveis. A representação resultante de uma palavra é dada pelo seu referente vetor na matriz de pesos de entrada W_x.

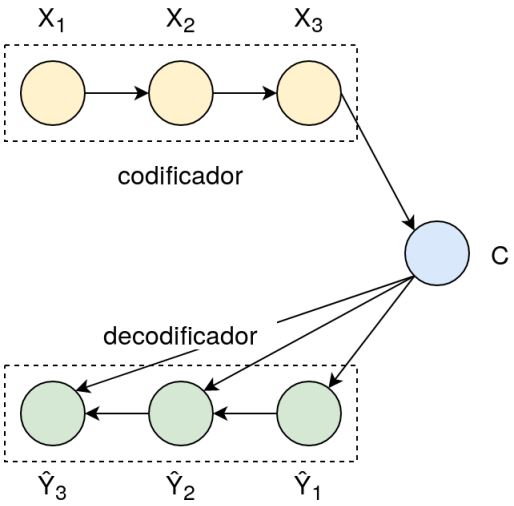


Figura 3.5: Arquitetura Encoder-Decoder, constituída por duas camadas de redes recorrentes, a de codificação, e a de decodificação, ligadas por um conjunto de pesos C que captura todo o contexto provindo da camada de codificação.

consiste em uma rede neural recorrente divida em duas etapas: a codificação, na qual uma sequência de tamanho variável é representada em um vetor; e a decodificação em que este mesmo vetor é utilizado para obter uma outra sequência de tamanho variável. O *Encoder-Decoder* é um modelo de aprendizado de sequências de tamanho variável a partir de outra sequência de tamanho variável como entrada, em que os tamanhos não necessariamente são os mesmos, reflexo do problema inicial em que o mesmo foi desenvolvido para atacar, a tradução de textos.

Essa estrutura toda é treinada conjuntamente e pode ser utilizada para gerar sequências de saída a partir de entradas ou avaliar um par de entrada e saída a partir da probabilidade $p_\theta(\mathbf{y} \mid \mathbf{x})$ aprendida pelo treinamento, no qual $\mathbf{x}$ e $\mathbf{y}$ representam respectivamente as sequências de entrada e saída e θ o conjunto de pesos do modelo treinado.

A etapa de codificação desse modelo pode ser vista como um resumo da sequência de entrada em um vetor de tamanho fixo, assim sendo, o mesmo pode ser utilizado para representações de documentos, ou palavras caso aplicada sequência de tamanho unitário. CHO *et al.* [51] demostram brevemente em seu trabalho que a codificação é capaz de capturar significados semânticos e sintático das palavras.

ELMo

ELMo [20], sigla para *Embeddings from Language Models*, em português Representações para Modelos Linguísticos, se diferencia do modelo de CHO *et al.* [51] por levar em consideração o contexto de uma dada palavra para obter sua representação, inspirado por PETERS *et al.* [52] e MCCANN *et al.* [53]. Isto é, uma palavra que

possua múltiplos sentidos terá representações diferentes dependendo da frase em que está inserida. O modelo é composto de uma rede LSTM [54] bi-direcional multicamada, treinada de maneira não supervisionada a partir do objetivo de prever a palavra seguinte.

Estudos anteriores mostram que em modelos de redes recorrentes multicamadas, a melhor camada escolhida para representação das palavras depende da finalidade em que se deseja aplicá-la [55] [56] [57] [58], e que em geral, camadas inferiores codificam melhor informação sintética e camadas superiores a informação semântica. A proposta de ELMo é para cada tarefa treinar uma combinação linear das representações obtida por cada camada da rede.

PETERS *et al.* [20] mostram que ELMo foi capaz de obter resultados melhores que o estado da arte em diversas tarefas do processamento de linguagem natural. Além de capturar informação sintética e semântica, ELMo se mostrou capaz de desambiguar o contexto de palavras, problema até então não resolvido pelos principais algoritmos de representação.

3.2.5 BERT

Outra técnica de representação bastante utilizada atualmente para representação de palavras é a BERT [21], *Bidirectional Transformers*, mas antes de explicá-la será apresentada abaixo a arquitetura Transformers em que a mesma se baseia.

Transformers

Um dos principais problemas de redes neurais recorrentes é o "esquecimento". O esquecimento é observado quando ao tentar prever uma palavra no final de uma longa sequência se perde influência das primeiras palavras da mesma por causa da sucessivas multiplicações dos pesos entre ambas. Apesar do algoritmo LSTM [54], *Long-Short Term Memory*, que visa atenuar esse efeito criando um peso adicional, chamado de portão, que controla o quanto os pesos de contexto são atualizados entre cada iteração de palavras de uma sequência, o esquecimento continua sendo um fator limitante.

O mecanismo de atenção desenvolvido por BAHDANAU *et al.* [59] também tem como objetivo diminuir o efeito do esquecimento. Este consiste em um conjunto de pesos adicionais que multiplicados pelos pesos de estados das iterações anteriores formam os pesos de estado da iteração corrente, como também descrito em mais detalhes por LUONG *et al.* [60]. A atenção se tornou uma das principais adições feitas a redes recorrentes aplicadas a processamento de linguagem natural.

Inspirado nesse mecanismo, VASWANI *et al.* [26] propuseram a arquitetura Transformer para tradução de textos. Essa arquitetura consiste em um sistema

de *Encoder-Decoder* desta vez composto por redes neurais *feed forward* multi camadas em que a dependência temporal entre as palavras de um documento é atacada apenas pelo sistema de *self-attention*, ou, atenção própria.

Enquanto o mecanismo de atenção é composto se dá por um único conjunto de pesos, a *self-attention* é dividido em 3: os pesos de busca, de resposta e de valor. Cada palavra do vocabulário terá um de cada vetor associado a mesma que será treinado em conjunto com o resto da arquitetura. Para cada palavra do documento será feito um produto interno do seu vetor de busca e os vetores de resposta das outras palavras da sentença, resultando em um valor único de atenção entre cada par possível de palavras de uma sentença. Após calcular esse valor será realizado uma operação de Softmax para que a norma do vetor de atenção seja unitária. Posteriormente se multiplica o vetor resultante pelo vetor de valor da palavra de resposta, essa operação tem a finalidade de diminuir a importância de palavras sem potencial discriminatório, como as *stopwords*. Assim são calculadas as entradas da rede *feed forward*.

A etapa de codificação do Transformer é composta por uma cascata de unidades formadas por uma camada de *self-attention* seguida de uma camada de rede *feed forward*. Diferentemente das redes recorrentes, o Transformer requer um tamanho fixo de entrada, sendo um parâmetro a ser definido a depender da base de treinamento escolhida. Apesar das redes *feed forward* não possuírem um conceito de sequência assim como as redes recorrentes, essa arquitetura provou ser eficiente em tarefas de processamento de linguagem natural ao mesmo tempo que possuem treinamento significativamente menos custosos que a mesma [26].

Arquitetura BERT

O sucesso do Transformer em traduções inspirou sua aplicação em outras tarefas. RADFORD *et al.* [61] propuseram utilizar apenas a camada de decodificação do Transformer junto com uma adaptação da sequência de entrada para adequar a tarefa a ser realizada. O modelo é pré-treinado com uma quantidade massiva de dados na predição da palavra seguinte e posteriormente é feito um ajuste fino com uma base de dados da tarefa desejada. Esta nova arquitetura foi capaz propagar os ganhos de performance do Transformer para toda gama de tarefas de NLP.

Entretanto, tanto o Transformer original quanto o de RADFORD *et al.* [61] são modelos unidirecionais, prevendo a palavra, ou sentença à direita a partir do contexto à esquerda. DEVLIN *et al.* [21] defendem que por serem unidirecionais os modelos restringem a capacidade dos mesmos, principalmente em tarefas que utilizam a sentença inteira, sendo a análise de sentimento um exemplo. Proporam então um modelo bidirecional praticamente idêntico ao Transformer de RADFORD *et al.* [61] sendo sua única modificação ter o mecanismo de atenção considerando o

contexto bidirecionalmente.

Assim como o ELMo, o BERT pode ser usado diretamente como classificador além de representar palavras. DEVLIN *et al.* [21] mostram as diferenças entre as camadas do modelo quando utilizados como representação, assim como nas arquiteturas apresentadas anteriormente as diferentes camadas possuem aprendizado de características distintas da língua.

3.3 Classificadores

Nessa seção serão abordadas as diferentes estratégias para classificação de sentimento. As etapas descritas anteriormente lidam com a preparação dos documentos para realização da classificação, apesar da complexidade dos métodos apresentados os classificadores podem ser tão simples quanto contadores de palavras positivas e negativas. Podemos dividi-los em algoritmos baseados em dicionário e algoritmos de aprendizado de máquina [6], suas características serão descritas abaixo.

3.3.1 Baseados em Dicionário

Uma das técnicas mais simples para classificação de sentimento é feita a partir da elaboração de um dicionário composto por palavras que tenham conotação positiva ou negativa. A formação de um dicionário de sentimento pode ser feita de forma manual [62] [63] ou de forma automática. Dentre as maneiras automáticas temos a aplicado por HU e LIU [64] que é feito a partir de um conjunto inicial de palavra anotadas e de um dicionário de sinônimos e antônimos tal qual o WordNet [65]. Um exemplo é selecionar as palavras semente "bom" e "ruim" e montar uma base de palavras recursivamente a partir dos seus sinônimos e antônimos. Formas similares de identificação de palavras com sentimento baseados em dicionários e palavras semente foram desenvolvidas por BLAIR-GOLDENSOHN *et al.* [66], RAO e RAVICHANDRAN [67], HASSAN e RADEV [68], entre outros.

Existindo um dicionário, a classificação de polaridade de um documento é dada com a partir de alguma função dos sentimentos das palavras que o compõem. É comum a inclusão algumas características sintáticas como identificação de adjetivos intensificadores e identificação de negação para ponderar a pontuação de sentimento de uma palavra ou expressão [6]. A função de classificação pode ser tão simples quanto uma média do sentimento das palavras ou, por exemplo, um calculo de proximidade entre o conjunto de palavras do documento e as palavras "excelente" e "ruim", representando as classes positivas e negativas, como propõe TURNEY [69].

3.3.2 Modelos Lineares

Naturalmente, as primeiras aplicações de aprendizado de máquina em análise de sentimento fizeram uso de modelos simples, como os modelos lineares. PANG *et al.* [14] fizeram um dos primeiros estudos comparativos entre estes métodos. A seguir, serão descritos os dois principais modelos lineares utilizados em processamento de linguagem natural: Naïve Bayes e Máquina de Vetor de Suporte.

Naïve Bayes

O classificador de Naïve Bayes se baseia no teorema de Bayes 3.1 e na premissa de independência entre as dimensões da representação dos dados. Quando representamos um documento textual com, por exemplo, *bag-of-words* a premissa de independência significaria assumir que existe independência entre as palavras do documento. Apesar de não existir independência entre palavras na linguagem, este modelo foi aplicado amplamente a tarefas de processamento de texto devido a sua performance e simplicidade.

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)} \tag{3.1}$$

SCHÜTZE *et al.* [70] descreveram a formulação matemática desse modelo, como será demostrado. Será denominado $\mathbf{X}$ o corpus de treinamento e $\mathbf{x}$ um documento desse corpus. As classes que os documentos pertencem, como positivo e negativo, é chamada de c_k, no qual $\mathbf{c}$ é o conjunto de classes. Portanto, substituindo as variáveis da equação 3.1 temos que a probabilidade de um documento pertencer a uma determinada classe é dada por:

$$p(c_k \mid \mathbf{x}) = \frac{p(c_k)\ p(\mathbf{x} \mid c_k)}{p(\mathbf{x})} \tag{3.2}$$

Como se assume independência entre as variáveis temos que:

$$p(x_i \mid x_{i+1}, \ldots, x_n, c_k) = p(x_i \mid c_k) \tag{3.3}$$

Logo, pode-se substituir $p(\mathbf{x} \mid c_k)$ pelo produtório de $p(x_i \mid c_k)$:

$$p(c_k \mid \mathbf{x}) = \frac{p(c_k) \prod_{i=1}^{n} p(x_i \mid c_k)}{p(\mathbf{x})} \tag{3.4}$$

A probabilidade $p(\mathbf{x})$ será constante, logo não influenciará o treinamento, portanto temos que a relação entre a probabilidade de um documento pertencer a classe é dada por:

$$p(c_k \mid \mathbf{x}) \propto p(c_k) \prod_{i=1}^{n} p(x_i \mid c_k) \tag{3.5}$$

Sendo assim, a tarefa da classificação que é encontrar a classe de maior probabilidade.

$$\hat{y} = \max p(c_k \mid \mathbf{x}) = \max_{k \in \{1,\ldots,K\}} p(c_k) \prod_{i=1}^{n} p(x_i \mid c_k) \tag{3.6}$$

A dependência entre a predição e o conjunto de treino então é dada por $p(c_k)$ e $p(\mathbf{x} \mid c_k)$, esses valores são obtidos por máxima verossimilhança. O parâmetro $p(c_k)$ é dado pela proporção de vezes que a classe aparece no conjunto de treino. Tendo o conjunto de treino o tamanho m de documentos:

$$\hat{p}(c_k) = \frac{\sum_{i=1}^{m} [y_i = c_k]}{m} \tag{3.7}$$

Por sua vez, $p(x_r \mid c_k)$ é estimado pela proporção entre uma determinada característica x_r com o total de de características do subconjunto de dados $\mathbf{X_{c_k}}$ pertencentes a classe c_k:

$$\hat{p}(x_r \mid c_k) = \frac{\sum_{j=i}^{m'} \sum_{i=1}^{n} [x_{ji} = x_r]}{|\mathbf{X_{c_k}}|} \tag{3.8}$$

O Naïve Bayes pode ser aplicado tanto a vetores inteiros da representação *bag-of-words* de frequência de palavras em um documento quanto pelo seu equivalente binário que assinala presença ou não de uma determinada palavra. Ao considerar a frequência de palavras aplicamos o modelo multinomial enquanto ao utilizar a representação de forma binária temos o modelo de Bernoulli. O modelo de Bernoulli é mais eficiente para documentos e vocabulários pequenos enquanto o multimodal se sobressaí no caso oposto [70].

Apesar da premissa de independência não ser verdadeira para língua, e não ser um bom estimador [70], Naïve Bayes se mostra eficiente na classificação, em especial quando existe um conjunto de características de semelhante importância. Sua principal vantagem se dá pelo baixo custo computacional de treinamento visto que seus parâmetros são estimados apenas por contagens na base de dados.

Máquina de Vetor de Suporte

A Máquina de Vetor de Suporte, também chamada de SVM, é um algoritmo que busca encontrar um vetor que melhor separe duas classes. A diferença entre esse algoritmo e outros classificadores como a regressão linear é que o SVM tem objetivo de obter o vetor que maximize a distância entre as margens das classes, enquanto a

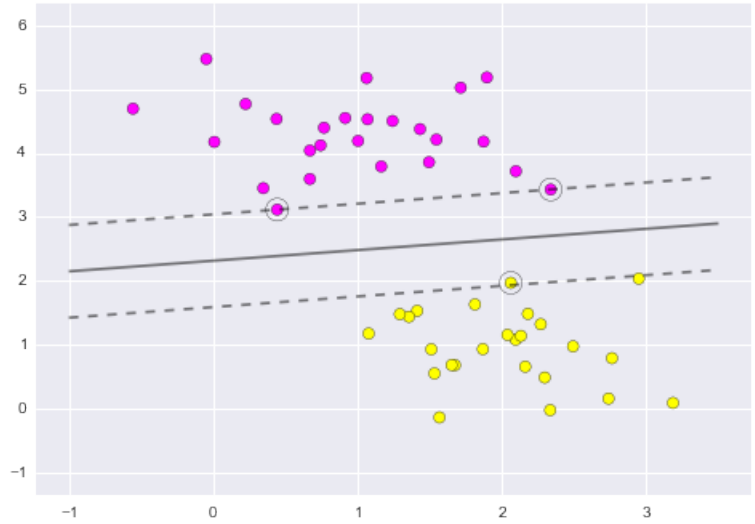


Figura 3.6: SVM classificando duas classes linearmente separáveis.
Imagem com direitos cedidos para uso não comercial, retirada de [71]

regressão linear, em geral, minimiza uma função custo baseada na distância entre o centro de massa das classes.

Entretanto, por se basear na margem das classes, o SVM não é capaz de separar classes que tenham sobreposição. Como forma de contornar esse problema foi criada uma variável de relaxamento de forma a controlar o quanto de sobreposição é permitido entre as classes como explicam CORTES e VAPNIK [72]. Ainda assim, até então a Máquina de Vetor de Suporte fica limitada a problemas lineares. Apenas com o desenvolvimento do chamado *kernel trick* que o SVM pode ser aplicável à problemas não-lineares e ganhou alta adoção. O *kernel trick* consiste em realizar uma transformação na representação dos dados de forma que os mesmos sejam linearmente separáveis após a transformação. Mapeamentos radiais e polinomiais são dois dos exemplos de *kernel* amplamente utilizados.

Uma das principais propriedades do SVM é que seu treinamento independe da dimensionalidade das características dos dados visto que seu aprendizado é feito com base nas margens entre as classes. Portanto, uma vez que os dados sejam separados por uma margem, mesmo que tenham alta dimensionalidade, o vetor de suporte de classificação pode ser encontrado. A aplicação de SVM para classificadores de texto foi proposta por JOACHIMS [73] visto que dados textuais representados por *bag-of-words* possuem 3 propriedades das quais se adéquam a classificação por SVM: *a*) possuem alta dimensionalidade *b*) são esparsos *c*) tem poucas dimensões irrelevantes. Apesar de JOACHIMS [73] demostrar a efetividade do SVM para classificação de texto utilizando *kernels* não lineares, bons resultados são obtidos mesmo sem a aplicação de *kernel* como mostra PANG *et al.* [14].

3.3.3 Modelos Não-Lineares

Classificadores lineares a partir de representação por *bag-of-words* ou TF-DF dos textos foram o estado da arte em tarefas de processamento natural durante os anos. Entretanto, como visto anteriormente, ao aplicar técnicas como *bag-of-words* abre-se mão do contexto em que cada palavra aparece no documento, este fator limita a performance da classificação. Sem conseguir capturar o contexto essas técnicas se distanciam de como nós interpretamos a língua. Para diminuir essa lacuna, possibilitados pelo desenvolvimento de técnicas de representações densas, passou a ser utilizados classificadores não-lineares que visam adicionar alguma forma de contexto no processo.

Deep Learning

Dentre os modelos não-lineares os que mais se destacam em processamento de linguagem natural são os de *Deep Learning*, baseada em redes neurais. Uma das principais características das redes neurais é que estas são algoritmos capazes de aproximar qualquer função [74]. Com o desenvolvimento do método de treinamento por auto diferenciação, chamado de *backpropagation*, nos anos 80 [75] as redes neurais foram amplamente adotadas. Entretanto, dois principais fatores inviabilizaram o treinamento de redes neurais de larga escala por muitos anos: o custo computacional do *backpropagation* e o efeito do *vanishing gradient* [76]. O *vanishing gradient* é decorrente do efeito multiplicativo do gradiente desde a camada de neurônios da saída até a camada de entrada. O mesmo resulta em um treinamento menor da camada a medida que a mesma está mais distantes da saída, estabelecendo assim uma dificuldade exponencial no treinamento com relação ao número de camadas.

Com o desenvolvimento tecnológico e a elaboração de uma técnica de treinamento camada-a-camada foi possível se sobrepor a esta barreira. Este treinamento, proposto por HINTON *et al.* [77], em que cada camada de neurônio é treinada individualmente e os valores obtidos por esse treinamento são utilizados como inicialização do treinamento da rede completa. Após esse trabalho começaram a testar redes neurais com um número cada vez maior de camadas. Denominou-se *Deep Learning* a esse conjunto de modelos de redes neurais profundas.

O salto de performance obtido pela aplicação de redes neurais profundas [22] em diversas áreas de conhecimento fez deste conjunto de técnicas um objeto de estudo de muito interesse por pesquisadores e indústria. Parte de seu sucesso se atribuí ao fato de que cada adicional na rede neural permite que a mesma obtenha uma representação mais complexa dos dados. Considerando o processamento de linguagem, diferentes tipos de redes neurais foram testados para tentar capturar a complexidade desta informação, com diferentes abordagens para adicionar o contexto em que

cada palavra se insere no processo de modelagem. As seções abaixo descrevem os principais tipos de redes neurais aplicados a NLP.

Redes Neurais Convolucionais

As redes convolucionais foram desenvolvidas prioritariamente para resolver problemas como a classificação de imagens e reconhecimento de fala. Um fator comum que essas tarefas compartilham é a possível translação da informação. Na classificação de imagens, por exemplo, um determinado objeto pode estar em diferentes posições ou ocupar tamanhos distintos em diferentes imagem. Apesar de ser possível aplicar redes neurais MLP ao problema, esta precisaria aprender os mesmos padrões de neurônios em diferentes posições [78]. Além disso, neste tipo de problema existe uma característica da localidade da informação, por exemplo, para reconhecimento de fala de uma palavra no final de uma frase, a palavra imediatamente anterior é, em geral, mais importante que a primeira palavra da mesma. Ao se utilizar redes neurais MLP completamente conectadas entre camadas se despreza essa propriedade, incorrendo em um maior custo computacional sem que necessariamente se reflita em performance, ou pior, aumentando a chance de o treinamento resultar em *overfitting*.

A forma da rede convolucional atacar este problema da correlação espacial da informação é utilizando filtros espaciais. Cada camada da rede convolucional é formada por um conjunto de filtros que são compostos por neurônios. A iteração de treinamento da rede consiste no deslocamento de cada filtro por toda dimensão da camada anterior. Desta maneira os padrões locais são capturados por um mesmo filtro independentemente de onde os padrões apareçam nos dados de entrada. O deslocamento dos filtros durante o treinamento funciona como um compartilhamento de pesos, reduzindo a quantidade de parâmetros livres da rede, e assim sua probabilidade de *overfitting*.

A interpretação de redes convolucionais aplicadas a processamento de texto não é muito diferente de sua utilização com imagens. Enquanto os filtros convolucionais exploram os padrões locais com filtros 2D de poucos pixels em imagens, com o caso textual os filtros são unidimensionais e percorrem a sequência de vetores de palavras com uma janela, capturando assim parte do contexto em que cada palavra se insere. Esse modelo já havia sido aplicado a documentos representados com *bag-of-words* como demostrado por KALCHBRENNER *et al.* [79] e YIH *et al.* [80] mas apenas quando utilizado em conjunto com a representação Word2Vec como apresentou KIM [30] que esse algoritmo ganhou tração nas tarefas de classificação de texto. A disponibilidade de modelos Word2Vec pré-treinadas, somado a representação densa que resulta em menor quantidade de parâmetros e menor probabilidade de *overfitting* torna essa combinação acessível para treinamento mesmo em casos de grandes bases

de dados, não obstante com performance superior como mostra KIM [30].

Redes Neurais Recorrentes

Outra forma de encapsular o contexto presente na linguagem é utilizando redes neurais recorrentes. As Redes Neurais Recorrentes são, em geral, treinadas com o algoritmo *Backpropagation Through Time* [81] (BPTT) em que a rede recorrente se comporta como uma rede neural *feedforward* na qual cada camada representa um passo temporal nos dados de entrada, no caso de texto podendo ser cada passo uma palavra. Portanto, ao aplicar esse algoritmo sobre o texto, o contexto sequencial do documento será capturado pela própria arquitetura da rede.

As variações de redes neurais recorrentes como apresentado em 3.2.4 em grande parte também realizam classificação. Um exemplo desta aplicação é mostrada por TAI *et al.* [82] classificando sentimento por uma rede LSTM. Posteriormente, técnicas de classificação que misturam diferentes tipos de redes foram testadas com êxito como mostram ZHOU *et al.* [31] em que a arquitetura é constituída de uma camada de rede recorrente seguida por uma camada de rede convolucional.

Apesar dos bons resultados obtidos nesses trabalhos, o treinamento por BPTT é computacionalmente mais custoso que o *backpropagation* de redes *feedforward* ou convolucionais, dado que o mesmo apresenta menos oportunidade de paralelismo do treinamento. Assim, esse algoritmo acaba sendo menos utilizado do que as redes convolucionais, ou até mesmo os Transformers para classificação de documentos.

Capítulo 4

Modelos de Redes Complexas

Grafos são estruturas de dados que codificam relações, e estão presentes em uma grande variedade de cenários. Estes são compostos por nós que representam elementos, quaisquer que sejam, e arestas que são as relações entre os elementos na rede. Se tratando de redes sociais, grafos são especialmente importantes dado que a relação entre os usuários é a principal componente desses serviços. Ao longo das ultimas décadas o estudo de modelos aplicados aos grafos, foi se tornando cada vez mais relevante. Esses modelos visam capturar diversas características possíveis das redes como a evolução do grafo no tempo, a propagação de epidemias dentro da rede, recomendação de arestas, a predição de classes de um dado nó ou aresta, entre outros.

Neste capitulo serão estudados tônicas de representações de vértices. Assim como no caso de documentos textuais, estes modelos, em geral, visam codificar a informação de um nó em um vetor de baixa dimensionalidade, capturando tanto seus atributos individuais quanto os decorrentes da estrutura de conexões do mesmo. Este objetivo se adéqua ao presente trabalho pois permite que seja treinado um classificador único que utilize tanto a informação capturada pela linguagem quanto da a provinda da rede.

4.1 Modelos Baseados em Fatoração Matricial

A primeira modalidade de técnicas de representação de nós a ser desenvolvida foi a por fatoração matricial. Essa consiste em utilizar o resultado da fatoração de uma das matrizes características da rede como representação dos nós, dentro os exemplos de matrizes dos grafos temos: a matriz de adjacência, a matriz Laplaciana, a matriz de centralidade de Katz, entre outras [83]. Nos casos que a matriz é positiva e semi-definida, como a matriz Laplaciana, é possível aplicar a decomposição em autovalores, nos outros casos, em geral, são utilizados métodos iterativos visando minimizar uma função custo.

Locally Linear Embedding (LLE)

ROWEIS e SAUL [84] desenvolveram esse algoritmo que tem como premissa que um nó é uma combinação linear de seus vizinhos. Portanto tendo a matriz de adjacência W_{ij} e o vetor de representação de um nó como Y_i, definimos o mesmo como:

$$Y_i \approx \sum_j W_{ij} Y_j \tag{4.1}$$

A função custo a ser minimizada para atingir é a que minimiza o erro de representação do somatório de todos os nós, sendo assim a função custo $\phi(Y)$ é dada por:

$$\phi(Y) = \sum_i |Y_i - \sum_j W_{ij} Y_j|^2 \tag{4.2}$$

Sendo necessário respeitar a restrição $\frac{1}{N} Y^T Y = 1$ para remover as soluções degeneradas. Esta técnica obtém uma representação de nós que preserva as distância de primeira ordem da rede, ou seja, vizinhos diretos no grafo original vão obter representações próximas.

Automapas Laplacianos

Esta técnica desenvolvida por BELKIN e NIYOGI [85] também tem como objetivo obter representações próximas entre vizinhos, desta vez inspirada na equação de fluxo de calor, usando a matriz laplaciana do grafo. Sua função custo é dada pela equação 4.3 na qual L é a laplaciana do grafo.

$$\begin{aligned} \phi(Y) &= \frac{1}{2} \sum_{ij} |Y_i - Y_j|^2 W_{ij} \\ &= tr(Y^T L Y) \end{aligned} \tag{4.3}$$

HOPE

Os algoritmos apresentados anteriormente são efetivos para manter a distância entre vizinhos de um grafo não-direcionado. Entretanto, grafos direcionados reais que em grande parte das vezes tem sua distribuição de grau seguindo uma lei de potência, como é o caso dos grafos formados em redes sociais, as propriedades de um par de nós não são simétricas. Para obter representações que respeitem essa assimetria OU *et al.* [86] desenvolveram o algoritmo *High-Order Proximity preserved Embedding* (HOPE). A chave deste algoritmo é utilizar métricas de proximidade de ordem superior em sua função custo, preservando assim as propriedades da assimetria. Qualquer medida de proximidade pode ser aplicada neste algoritmo, OU *et al.* [86] em seu estudo

testaram medidas como a proximidade de Katz e Pagerank personalizado, mostrando que em vários casos essas medidas possuem aproximações que diminuem o custo computacional da otimização.

HOPE representa cada vértice em 2 vetores, um de entrada e um de saída, o qual denominaremos respectivamente Y^s e Y^t. A função custo é dada pela equação 4.4 na qual S representa a matriz de proximidade entre os nós dada por qualquer uma das medidas escolhidas.

$$\phi = ||S - Y^s Y^{t^T}||_F^2 \tag{4.4}$$

4.2 Modelos Baseados em Passeios Aleatórios

Outra estratégia amplamente adotada são as baseadas em Passeios Aleatórios. O Passeio Aleatório é um processo estocástico que consiste em se selecionar um vértice inicial no grafo e completar uma sequência de passos aleatórios. Sua aplicação para obtenção de representações considera a sequência de nós de múltiplas realizações do Passeio Aleatório como uma função de proximidade entre os nós. Por ser iterativo e não necessitar o calculo de nenhuma matriz completa do grafo, esse método apresenta vantagem em custo operacional para redes grandes. Além disso, a iteratividade do método também permite que novos elementos sejam adicionados ao grafo sem requerer um treinamento completo do modelo, característica essencial para seleção de algoritmos em diversas aplicações. A seguir serão apresentados os principais modelos deste grupo.

Deepwalk

Deepwalk [87] se inspira em modelos de representações de palavras utilizando como entrada sequências de iterações de Passeios Aleatórios truncadas em poucos passos. Os autores observaram que assim como a distribuição de ocorrência de palavras de uma língua segue uma lei de potência, mesmo comportamento de boa parte das redes formadas no mundo real, como a de conexão entre pessoas. Desta forma, adaptaram algoritmos previamente aplicados para predição de palavras para obter as representações dos vértices.

O modelo proposto é muito similar a variação *skipgram* do Word2Vec 3.2.3, os passeios aleatórios truncados formam sequências de tamanho $2k + 1$, o treinamento do algoritmo é feito de maneira a maximizar a probabilidade de predição do contexto a partir do vértice na posição central do passeio. A formula 4.5 formaliza o problema citado dado que v_i é o vértice central de um dado passeio e Φ é o mapeamento de um vetor em sua representação, que queremos encontrar.

$$\max P(\{v_{i-k}, ..., v_{i-1}, v_{i+1}, ..., v_{i+k}\}|\Phi(v_i)) \tag{4.5}$$

O processo é feito de forma iterativa em que cada realização do Passeio Aleatório compões uma entrada da algoritmo. A otimização é dada por gradiente descendente da função custo ϕ que maximiza a probabilidade 4.5, esta é apresentada na equação 4.6. Outras adaptações propostas posteriormente ao Word2Vec para diminuir o custo computacional de treinamento também podem ser adotadas no Deepwalk, como *softmax* hierárquico e *negative sampling*.

$$\phi = -\log P(\{v_{i-k}, ..., v_{i-1}, v_{i+1}, ..., v_{i+k}\}|\Phi(v_i)) \tag{4.6}$$

node2vec

O node2vec [88] também é um algoritmo inspirado no Word2Vec. Seu processo de otimização é idêntico ao apresentado pelo Deepwalk, também aplicado sobre sequências de passeios aleatórios de tamanho fixo. A diferença entre ambos está na estratégia de passeio aleatório, que além de capturar a vizinhança de um dado vértice visa também capturar a posição do mesmo na estrutura ao seu redor. Enquanto a vizinhança de um nó compõe uma parte de sua informação, o posicionamento do mesmo em sua comunidade também é relevante. Dentro de uma rede, um nó central de uma comunidade pode apresentar características mais semelhantes ao ponto central de outra comunidade do que a algum vértice vizinho com poucas conexões.

A proposta desenvolvida por GROVER e LESKOVEC [88] consiste em um Passeio Aleatório enviesado que parametrizam quão relevante são cada um desses dois comportamentos. Ao se escolher parâmetros que priorizem a vizinhança do vértice, a amostragem do Passeio Aleatório enviesado passa a se assemelhar a uma busca em profundidade, por outro lado, focando-se na estrutura formada pelas conexões do nó obtemos uma amostragem semelhante a busca em largura.

No Passeio Aleatório não enviesado, a probabilidade de transição π_{uv} do nó u para o nó v é dada por $\pi_{uv} = \frac{w_{uv}}{\sum w_u}$, no qual w_{uv} é o peso da aresta entre u e v e $\sum w_u$ é o somatório dos das arestas de u, neste caso operando como regularizador da probabilidade. Para atingir seu objetivo, node2vec multiplica a probabilidade de transição original do Passeio Aleatório por um fator $\alpha(s, v)$ parametrizado pelos parâmetros p e q, como mostra a equação 4.7, na qual s representa o vértice em que o Passeio Aleatório amostrou no passo anterior e d_{sv} representa a distância mínima entre o par de vértices s e v. Esses parâmetros controlam quão rápido o Passeio Aleatório explora a rede.

$$\alpha(s,v) = \begin{cases} \frac{1}{p}, & \text{se } d_{sv} = 0 \\ 1, & \text{se } d_{sv} = 1 \\ \frac{1}{q}, & \text{se } d_{sv} = 2 \end{cases} \tag{4.7}$$

O parâmetro p é chamado de parâmetro de retorno, a escolha de valores altos para esse fator faz com que o Passeio Aleatório evite re-amostrar vértices recém amostrados, enquanto valores baixos forçam o algoritmo a manter a amostragem localizada em uma distância pequena de seu ponto de partida. Por sua vez, o parâmetro q regula a probabilidade de se amostrar nós distantes do vértice de partida. Valores baixos de q resultam em uma amostragem semelhante a busca em profundidade enquanto valores altos levam a um padrão de busca em largura. Resumidamente, a probabilidade de transição do passeio aleatório proposto por node2vec é dada pela equação 4.8.

$$\pi(s,u,v) = \frac{w_{uv}\alpha(s,v)}{\sum_x w_{ux}\alpha(s,x)} \tag{4.8}$$

Os autores mostram que a adoção dessa estratégia de amostragem resultou em ganhos significativos de performance do algoritmo para tarefas de classificação de vértices. Apesar da maior complexidade, o algoritmo apresenta aumento linear do custo computacional com relação ao número de nós da rede, possibilitando sua aplicação em redes de grande volume.

4.3 Redes Convolucionais de Grafos

Adaptações de modelos de *Deep Learning* também passaram a ser utilizados para representação de nós, como os modelos por *autoencodes* [89] e [90]. A rede convolucional de grafos [91] (GCN) é um dos principais algoritmos deste grupo, no qual se desenvolveu uma adaptação do operador de convolução para aplicar em grafos.

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^l W^l) \tag{4.9}$$

A equação 4.9 define a camada de convolução usada nesse algoritmo. A matriz $\tilde{A} = A + I$ é a matriz de adjacência somada a matriz identidade, correspondente a um grafo com um laço [1] adicionado a cada vértice, A matriz diagonal $\tilde{D}$ é composta por $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, H^l representa a saída da camada l e W^l os pesos treináveis da mesma, por fim, temos a função de ativação escolhida σ. Os autores mostram em seu trabalho a relação desta camada com a aproximação de primeira ordem da convolução espectral do grafo.

[1] aresta conectando um vértice nele mesmo

Apesar do GCN não ser especificamente voltado para obtenção de representações, trabalhos posteriores realizaram adaptações no algoritmo com essa finalidade. Outra dificuldade é que assim como os modelos de fatoração matricial, só é possível aplicá-lo em grafos fixos, ao precisar adicionar novos vértices precisa-se retreinar o modelo. Dentre os algoritmos que se propõe a atacar esses problemas, destacamos o GraphSAGE [92], que minimiza a representação de um vértice a partir de uma função de agregação de seus vizinhos, desta forma a técnica não depende da matriz de adjacência completa para seu treinamento.

Capítulo 5

Método

Capítulo 6

Resultados e Discussões

Capítulo 7

Conclusões

Referências Bibliográficas

[1] GO, A., BHAYANI, R., HUANG, L. "Twitter sentiment classification using distant supervision", *CS224N Project Report, Stanford*, v. 1, n. 12, pp. 2009, 2009.

[2] SOCIAL, W. A. "Digital in 2019: Global Overview". `https://wearesocial.com/global-digital-report-2019`. acessado em 3 de Junho de 2019.

[3] TURING, A. M. "Computing machinery and intelligence", *Mind*, v. 59, n. 236, pp. 433–460, 1950.

[4] LIU, B. *Opinions, Sentiment, and Emotion in Text*. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press, 2015. ISBN: 9781107017894. Disponível em: <`https://books.google.com.br/books?id=6IdsCQAAQBAJ`>.

[5] CAMBRIA, E. "Affective computing and sentiment analysis", *IEEE Intelligent Systems*, v. 31, n. 2, pp. 102–107, 2016.

[6] TABOADA, M., BROOKE, J., TOFILOSKI, M., et al. "Lexicon-based methods for sentiment analysis", *Computational linguistics*, v. 37, n. 2, pp. 267–307, 2011.

[7] DAS, S. R., CHEN, M. Y. "Yahoo! for Amazon: Sentiment extraction from small talk on the web", *Management science*, v. 53, n. 9, pp. 1375–1388, 2007.

[8] RILOFF, E., WIEBE, J., PHILLIPS, W. "Exploiting subjectivity classification to improve information extraction". In: *AAAI*, pp. 1106–1111, 2005.

[9] SOCHER, R., PENNINGTON, J., HUANG, E. H., et al. "Semi-supervised recursive autoencoders for predicting sentiment distributions". In: *Proceedings of the conference on empirical methods in natural language processing*, pp. 151–161. Association for Computational Linguistics, 2011.

[10] SOCHER, R., PERELYGIN, A., WU, J., et al. "Recursive deep models for semantic compositionality over a sentiment treebank". In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

[11] NASUKAWA, T., YI, J. "Sentiment analysis: Capturing favorability using natural language processing". In: *Proceedings of the 2nd international conference on Knowledge capture*, pp. 70–77. ACM, 2003.

[12] SNYDER, B., BARZILAY, R. "Multiple aspect ranking using the good grief algorithm". In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 300–307, 2007.

[13] TWITTER. "Q2-2019 Letter to Shareholders". `https://s22.q4cdn.com/826641620/files/doc_financials/2019/q2/Q2-2019-Shareholder-Letter.pdf`. acessado em 07 de Novembro de 2019.

[14] PANG, B., LEE, L., VAITHYANATHAN, S. "Thumbs up?: sentiment classification using machine learning techniques". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86. Association for Computational Linguistics, 2002.

[15] READ, J. "Using emoticons to reduce dependency in machine learning techniques for sentiment classification". In: *Proceedings of the ACL student research workshop*, pp. 43–48. Association for Computational Linguistics, 2005.

[16] WANG, S., MANNING, C. D. "Baselines and bigrams: Simple, good sentiment and topic classification". In: *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*, pp. 90–94. Association for Computational Linguistics, 2012.

[17] PALTOGLOU, G., THELWALL, M. "A study of information retrieval weighting schemes for sentiment analysis". In: *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 1386–1395. Association for Computational Linguistics, 2010.

[18] MIKOLOV, T., SUTSKEVER, I., CHEN, K., et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[19] PENNINGTON, J., SOCHER, R., MANNING, C. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[20] PETERS, M. E., NEUMANN, M., IYYER, M., et al. "Deep contextualized word representations". In: *Proc. of NAACL*, 2018.

[21] DEVLIN, J., CHANG, M.-W., LEE, K., et al. "Bert: Pre-training of deep bidirectional transformers for language understanding", *arXiv preprint arXiv:1810.04805*, 2018.

[22] LECUN, Y., BENGIO, Y., HINTON, G. "Deep learning", *nature*, v. 521, n. 7553, pp. 436–444, 2015.

[23] KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[24] HINTON, G., DENG, L., YU, D., et al. "Deep neural networks for acoustic modeling in speech recognition", *IEEE Signal processing magazine*, v. 29, 2012.

[25] ESTEVA, A., KUPREL, B., NOVOA, R. A., et al. "Dermatologist-level classification of skin cancer with deep neural networks", *Nature*, v. 542, n. 7639, pp. 115, 2017.

[26] VASWANI, A., SHAZEER, N., PARMAR, N., et al. "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[27] GE, T., WEI, F., ZHOU, M. "Reaching human-level performance in automatic grammatical error correction: An empirical study", *arXiv preprint arXiv:1807.01270*, 2018.

[28] AKBIK, A., BLYTHE, D., VOLLGRAF, R. "Contextual string embeddings for sequence labeling". In: *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1638–1649, 2018.

[29] WU, Y., HU, B. "Learning to extract coherent summary via deep reinforcement learning". In: *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[30] KIM, Y. "Convolutional Neural Networks for Sentence Classification". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A*

meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1746–1751, 2014. Disponível em: <http://aclweb.org/anthology/D/D14/D14-1181.pdf>.

[31] ZHOU, P., QI, Z., ZHENG, S., et al. "Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling", *arXiv preprint arXiv:1611.06639*, 2016.

[32] ALBERT, R., BARABÁSI, A.-L. "Statistical mechanics of complex networks", *Rev. Mod. Phys.*, v. 74, pp. 47–97, Jan 2002. doi: 10.1103/RevModPhys.74.47. Disponível em: <https://link.aps.org/doi/10.1103/RevModPhys.74.47>.

[33] BARABÁSI, A.-L., GULBAHCE, N., LOSCALZO, J. "Network medicine: a network-based approach to human disease", *Nature reviews genetics*, v. 12, n. 1, pp. 56, 2011.

[34] HUFNAGEL, L., BROCKMANN, D., GEISEL, T. "Forecast and control of epidemics in a globalized world", *Proceedings of the National Academy of Sciences*, v. 101, n. 42, pp. 15124–15129, 2004.

[35] ALBERT, R. "Attack and error tolerance in complex networks", *Nature*, v. 406, pp. 387–482, 2000.

[36] COLLADON, A. F., REMONDI, E. "Using social network analysis to prevent money laundering", *Expert Systems with Applications*, v. 67, pp. 49–58, 2017.

[37] RATKIEWICZ, J., CONOVER, M. D., MEISS, M., et al. "Detecting and tracking political abuse in social media". In: *Fifth international AAAI conference on weblogs and social media*, 2011.

[38] VAROL, O., DAVIS, C. A., MENCZER, F., et al. "Feature engineering for social bot detection", *Feature engineering for machine learning and data analytics*, p. 311, 2018.

[39] BACKSTROM, L., LESKOVEC, J. "Supervised random walks: predicting and recommending links in social networks". In: *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 635–644. ACM, 2011.

[40] QIU, J., TANG, J., MA, H., et al. "Deepinf: Social influence prediction with deep learning". In: *Proceedings of the 24th ACM SIGKDD Internatio-*

nal Conference on Knowledge Discovery & Data Mining, pp. 2110–2119. ACM, 2018.

[41] KISS, T., STRUNK, J. "Unsupervised multilingual sentence boundary detection", *Computational Linguistics*, v. 32, n. 4, pp. 485–525, 2006.

[42] DAMERAU, F. J. "A technique for computer detection and correction of spelling errors", *Communications of the ACM*, v. 7, n. 3, pp. 171–176, 1964.

[43] NAVARRO, G. "A guided tour to approximate string matching", *ACM computing surveys (CSUR)*, v. 33, n. 1, pp. 31–88, 2001.

[44] PORTER, M. F. "An algorithm for suffix stripping", *program*, v. 14, n. 3, pp. 130–137, 1980.

[45] LO, R. T.-W., HE, B., OUNIS, I. "Automatically building a stopword list for an information retrieval system". In: *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR)*, v. 5, pp. 17–24, 2005.

[46] SAIF, H., FERNÁNDEZ, M., HE, Y., et al. "On stopwords, filtering and data sparsity for sentiment analysis of twitter", 2014.

[47] MANNING, C., RAGHAVAN, P., SCHÜTZE, H. "Introduction to information retrieval", *Natural Language Engineering*, v. 16, n. 1, pp. 100–103, 2010.

[48] POWERS, D. M. "Applications and explanations of Zipf's law". In: *Proceedings of the joint conferences on new methods in language processing and computational natural language learning*, pp. 151–160. Association for Computational Linguistics, 1998.

[49] SALTON, G., BUCKLEY, C. "Term-weighting approaches in automatic text retrieval", *Information processing & management*, v. 24, n. 5, pp. 513–523, 1988.

[50] BOJANOWSKI, P., GRAVE, E., JOULIN, A., et al. "Enriching word vectors with subword information", *Transactions of the Association for Computational Linguistics*, v. 5, pp. 135–146, 2017.

[51] CHO, K., VAN MERRIËNBOER, B., GULCEHRE, C., et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation", *arXiv preprint arXiv:1406.1078*, 2014.

[52] PETERS, M. E., AMMAR, W., BHAGAVATULA, C., et al. "Semi-supervised sequence tagging with bidirectional language models", *arXiv preprint arXiv:1705.00108*, 2017.

[53] MCCANN, B., BRADBURY, J., XIONG, C., et al. "Learned in translation: Contextualized word vectors". In: *Advances in Neural Information Processing Systems*, pp. 6294–6305, 2017.

[54] HOCHREITER, S., SCHMIDHUBER, J. "Long short-term memory", *Neural computation*, v. 9, n. 8, pp. 1735–1780, 1997.

[55] HASHIMOTO, K., XIONG, C., TSURUOKA, Y., et al. "A joint many-task model: Growing a neural network for multiple nlp tasks", *arXiv preprint arXiv:1611.01587*, 2016.

[56] SØGAARD, A., GOLDBERG, Y. "Deep multi-task learning with low level tasks supervised at lower layers". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 231–235, 2016.

[57] BELINKOV, Y., DURRANI, N., DALVI, F., et al. "What do neural machine translation models learn about morphology?" *arXiv preprint arXiv:1704.03471*, 2017.

[58] MELAMUD, O., GOLDBERGER, J., DAGAN, I. "context2vec: Learning generic context embedding with bidirectional lstm". In: *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pp. 51–61, 2016.

[59] BAHDANAU, D., CHO, K., BENGIO, Y. "Neural machine translation by jointly learning to align and translate", *arXiv preprint arXiv:1409.0473*, 2014.

[60] LUONG, M.-T., PHAM, H., MANNING, C. D. "Effective approaches to attention-based neural machine translation", *arXiv preprint arXiv:1508.04025*, 2015.

[61] RADFORD, A., NARASIMHAN, K., SALIMANS, T., et al. "Improving language understanding by generative pre-training", Disponível em: <https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf>.

[62] STONE, P. J., DUNPHY, D. C., SMITH, M. S. "The general inquirer: A computer approach to content analysis." 1966.

[63] TONG, R. M. "An operational system for detecting and tracking opinions in on-line discussion". In: *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification*, v. 1, 2001.

[64] HU, M., LIU, B. "Mining and summarizing customer reviews". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177. ACM, 2004.

[65] MILLER, G. A., BECKWITH, R., FELLBAUM, C., et al. "Introduction to WordNet: An on-line lexical database", *International journal of lexicography*, v. 3, n. 4, pp. 235–244, 1990.

[66] BLAIR-GOLDENSOHN, S., HANNAN, K., MCDONALD, R., et al. "Building a sentiment summarizer for local service reviews". In: *Proceedings of WWW-2008 workshop on NLP in the Information Explosion Era*, 2008.

[67] RAO, D., RAVICHANDRAN, D. "Semi-supervised polarity lexicon induction". In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 675–682. Association for Computational Linguistics, 2009.

[68] HASSAN, A., RADEV, D. "Identifying text polarity using random walks". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 395–403. Association for Computational Linguistics, 2010.

[69] TURNEY, P. D. "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews". In: *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 417–424. Association for Computational Linguistics, 2002.

[70] SCHÜTZE, H., MANNING, C. D., RAGHAVAN, P. "Introduction to information retrieval". In: *Proceedings of the international communication of association for computing machinery conference*, v. 4, 2008.

[71] VANDERPLAS, J. `https://github.com/jakevdp/sklearn_pycon2015/blob/master/notebooks/03.1-Classification-SVMs.ipynb`. acessado em 28 de Maio 2017.

[72] CORTES, C., VAPNIK, V. "Support-vector networks", *Machine learning*, v. 20, n. 3, pp. 273–297, 1995.

[73] JOACHIMS, T. "Text categorization with support vector machines: Learning with many relevant features". In: *European conference on machine learning*, pp. 137–142. Springer, 1998.

[74] HORNIK, K., STINCHCOMBE, M., WHITE, H. "Multilayer feedforward networks are universal approximators", *Neural networks*, v. 2, n. 5, pp. 359–366, 1989.

[75] WERBOS, P. J. "Applications of advances in nonlinear sensitivity analysis". In: *System modeling and optimization*, Springer, pp. 762–770, 1982.

[76] HOCHREITER, S. "The vanishing gradient problem during learning recurrent neural nets and problem solutions", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, v. 6, n. 02, pp. 107–116, 1998.

[77] HINTON, G. E., OSINDERO, S., TEH, Y.-W. "A fast learning algorithm for deep belief nets", *Neural computation*, v. 18, n. 7, pp. 1527–1554, 2006.

[78] LECUN, Y., BENGIO, Y., OTHERS. "Convolutional networks for images, speech, and time series", *The handbook of brain theory and neural networks*, v. 3361, n. 10, 1995.

[79] KALCHBRENNER, N., GREFENSTETTE, E., BLUNSOM, P. "A convolutional neural network for modelling sentences", *arXiv preprint arXiv:1404.2188*, 2014.

[80] YIH, W.-T., HE, X., MEEK, C. "Semantic parsing for single-relation question answering". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 643–648, 2014.

[81] WILLIAMS, R. J., ZIPSER, D. "Gradient-based learning algorithms for recurrent", *Backpropagation: Theory, architectures, and applications*, v. 433, 1995.

[82] TAI, K. S., SOCHER, R., MANNING, C. D. "Improved semantic representations from tree-structured long short-term memory networks", *arXiv preprint arXiv:1503.00075*, 2015.

[83] GOYAL, P., FERRARA, E. "Graph embedding techniques, applications, and performance: A survey", *Knowledge-Based Systems*, v. 151, pp. 78–94, 2018.

[84] ROWEIS, S. T., SAUL, L. K. "Nonlinear dimensionality reduction by locally linear embedding", *science*, v. 290, n. 5500, pp. 2323–2326, 2000.

[85] BELKIN, M., NIYOGI, P. "Laplacian eigenmaps and spectral techniques for embedding and clustering". In: *Advances in neural information processing systems*, pp. 585–591, 2002.

[86] OU, M., CUI, P., PEI, J., et al. "Asymmetric transitivity preserving graph embedding". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1105–1114, 2016.

[87] PEROZZI, B., AL-RFOU, R., SKIENA, S. "Deepwalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

[88] GROVER, A., LESKOVEC, J. "node2vec: Scalable feature learning for networks". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

[89] WANG, D., CUI, P., ZHU, W. "Structural deep network embedding". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1225–1234, 2016.

[90] CAO, S., LU, W., XU, Q. "Deep neural networks for learning graph representations". In: *Thirtieth AAAI conference on artificial intelligence*, 2016.

[91] KIPF, T. N., WELLING, M. "Semi-supervised classification with graph convolutional networks", *arXiv preprint arXiv:1609.02907*, 2016.

[92] HAMILTON, W., YING, Z., LESKOVEC, J. "Inductive representation learning on large graphs". In: *Advances in neural information processing systems*, pp. 1024–1034, 2017.