APÊNDICE I – Principais algoritmos de Aprendizagem de Máquina

Aprendizagem Supervisionada

Regressão Linear	Os algoritmos de regressão linear mostram ou preveem a relação entre duas variáveis ou dois fatores ajustando uma linha reta contínua aos dados. A linha geralmente é calculada com a função Custo de Erro Quadrado. A regressão linear é um dos tipos mais populares de análise de regressão.
Regressão Logística	Os algoritmos de regressão logística ajustam uma curva contínua em forma de S aos dados. A regressão logística é outro tipo popular de análise de regressão, embora seja utilizada para tarefas de classificação.
K-Vizinhos mais próximos (K-Nearest Neighbors)	Os algoritmos de K-vizinho mais próximos armazenam todos os pontos de dados disponíveis e classificam cada novo ponto de dados com base nos pontos de dados mais próximos a eles, conforme medido por uma função de distância.
Máquinas de Vetores de Suporte (SVMs)	Os algoritmos de máquinas de vetores de suporte desenham um hiperplano entre os dois pontos de dados mais próximos. Isso diferencia as classes e maximiza as distâncias entre elas para diferenciá-las mais claramente.
Árvores de Decisão (Decision Trees)	Os algoritmos de árvore de decisão dividem os dados em dois ou mais conjuntos homogêneos. Eles usam as regras se–então para separar os dados com base no diferenciador mais significativo entre os pontos de dados.
Florestas Aleatórias (Random Forests)	Os algoritmos de floresta aleatória se baseiam nas árvores de decisão, mas em vez de criar uma árvore, eles criam uma floresta de árvores. Em seguida, agregam votos de diferentes formações aleatórias das árvores de decisão para determinar a classe final do objeto de teste.
Redes Neurais	

Aprendizagem Não Supervisionada

K-Means	Os algoritmos K-means classificam os dados em grupos (clusters) – em que K é igual ao número de clusters. Os pontos de dados dentro de cada cluster são homogêneos e são heterogêneos para pontos de dados em outros clusters.
DBSCAN	Com base em um conjunto de pontos, o DBSCAN agrupa pontos que estão próximos uns dos outros com base em uma medição de distância (geralmente distância euclidiana) e um número mínimo de pontos. Ele também marca como outliers os pontos que estão em regiões de baixa densidade.
Análise de Agrupamentos Hierárquicos (HCA)	Os algoritmos HCA usam uma estratégia que busca construir uma hierarquia de clusters que tenha uma ordenação estabelecida de cima para baixo.
Análise de Componente Principal (PCA)	Os algoritmos de análise de componente principal ou PCA são utilizados para redução de dimensionalidade (campos) de grandes arquivos de dados. Reduzir o número de campos torna mais simples explorar e visualizar grandes arquivos de dados.

Aprendizagem por Reforço

REINFORCE	Algoritmo clássico de 1992 que deu visibilidade à área de aprendizado por força, REINFORCE é uma variante de Monte Carlo de um algoritmo de gradiente de política (policy) no aprendizado por reforço. O agente coleta amostras de um episódio usando sua política (policy) atual e as usa para atualizar o parâmetro θ da política. Uma vez que uma trajetória completa deve

	ser concluída para construir um espaço de amostra, ela é atualizada como um algoritmo off-policy.
SARSA	Sarsa usa diferença temporal para aprendizagem (*TD-learning*), que combina métodos de Monte Carlo, para aprender direto da experiência obtida pelo agente sem um modelo representativo das dinâmicas do ambiente, com o uso de programação dinâmica.
Deep Q-Learning (DQL)	O objetivo do Q-Learning é aprender uma política que diz a um agente que ação tomar sob determinadas circunstâncias (estados). O DQL estende e adapta a ideia por trás do Q-Learning para uso com redes neurais como aproximadores universais para representar problemas de alta dimensionalidade.
Rainbow	Várias extensões foram feitas aos algoritmos de DQL e o Rainbow combina 6 das técnicas mais proeminentes em um único algoritmo com ótimos resultados.
Deep Deterministic Policy Gradient (DDPG)	DDPG é um algoritmo off-policy, actor-critic, em que são combinadas duas redes neurais especializadas. Uma das redes neurais é o ator que prevê a ação a ser realizada e a outra rede, o crítico, avalia a performance do ator para indicar qual o melhor caminha de maximizar a recompensa.
Proximal Policy Optimization Algorithms (PPO)	PPO é um algoritmo on-policy de otimização de primeira ordem que busca políticas (policies) mais eficientes e compara com a corrente para troca por vários ciclos.
Asynchronous Advantage Actor-Critic (A3C)	O A3C é um algoritmo que usa múltiplos atores experimentando o ambiente em paralelo e, com isso, torna os dados em um processo mais estacionário e também torna a exploração dos estados mais diversa.
Twin Delayed Deep Deterministic policy gradient (TD3)	O TD3 baseia-se no algoritmo DDPG, com algumas modificações destinadas a lidar com o viés de superestimação com a função de valor. Em particular, ele utiliza clipped double Q-learning, atualização menos frequente de redes alvo (target) e política (policy) e suavização de política alvo (target policy) (semelhante a uma atualização baseada em SARSA. Uma atualização mais estável).
AlphaZero	AlphaZero é um algoritmo de aprendizado por reforço para jogos de tabuleiro como Go, xadrez e shogi. Ele ganhou grande visibilidade em 2017 ao atingir níveis sobre-humanos nós três jogos citados. Ele é um marco para o Go, pois é o primeiro algoritmo a conseguir vencer campeões mundiais do jogo. O AlphaZero trouxe enorme interesse da comunidade de IA para o aprendizado por reforço por usar IA para resolver problemas que seriam computacionalmente intratáveis de maneira exata com algoritmos tradicionais.
MuZero	O MuZero combina a busca utilizada no AlphaZero (Monte Carlo Tree Search – MCTS) com um modelo aprendido das dinâmicas do ambiente (environment). Ele atinge o nível sobre-humano em 51 dos 57 jogos do videogame Atari, que é muito usado para *benchmarking* por envolver tarefas variadas e com vários níveis de complexidade e fácil obtenção de recompensas (ex.: pontuação nos jogos).
Agent57	O Agent57 é um algoritmo bem recente, 2020, e é o primeiro algoritmo de aprendizado por reforço a atingir nível sobre-humano em todos os 57 jogos. Ele é *model-free*, ao contrário do MuZero, e possui um meta-controlador que possui um mecanismo adaptativo para escolher qual política (policy) deve fazer a exploração do espaço e qual deve fazer o *exploiting* além do desmembramento da recompensa entre extrínseca e intrínseca para melhor estabilidade do treinamento.

APÊNDICE II – Redes Neurais Artificiais[1]

0. Algoritmos tradicionais de aprendizagem de máquina dependem fortemente da representação dos dados para serem capazes de criar relacionamentos entre tais dados e as predições a que eles podem conduzir. Note-se, por exemplo, a diferença entre um sistema de diagnóstico que depende de informações de um paciente fornecidas pelo médico (ex.: IMC, tipo sanguíneo, glicemia) para ser capaz de propor um diagnóstico, de um sistema capaz de identificar tumores a partir de uma imagem radiográfica. Enquanto algoritmos tradicionais são capazes de extrair correlações entre o primeiro grupo de informações informadas pelo médico – aqui referidas como *features* ou atributos - e um possível diagnóstico, no segundo exemplo tais sistemas possuem limitações na análise de dados não estruturados como imagens, pois não são capazes de extrair significado a partir de tão somente um conjunto de pixels.

1. Uma solução para esse tipo de problema é o uso de técnicas que sejam capazes não somente de aprenderem relacionamentos entre os atributos e a resposta ou saída (predição), mas também de aprenderem a melhor forma de representar os dados de entrada. É nesse contexto que se insere a técnica de *deep learning*, que permite expressar representações complexas em termos de outras mais simples.

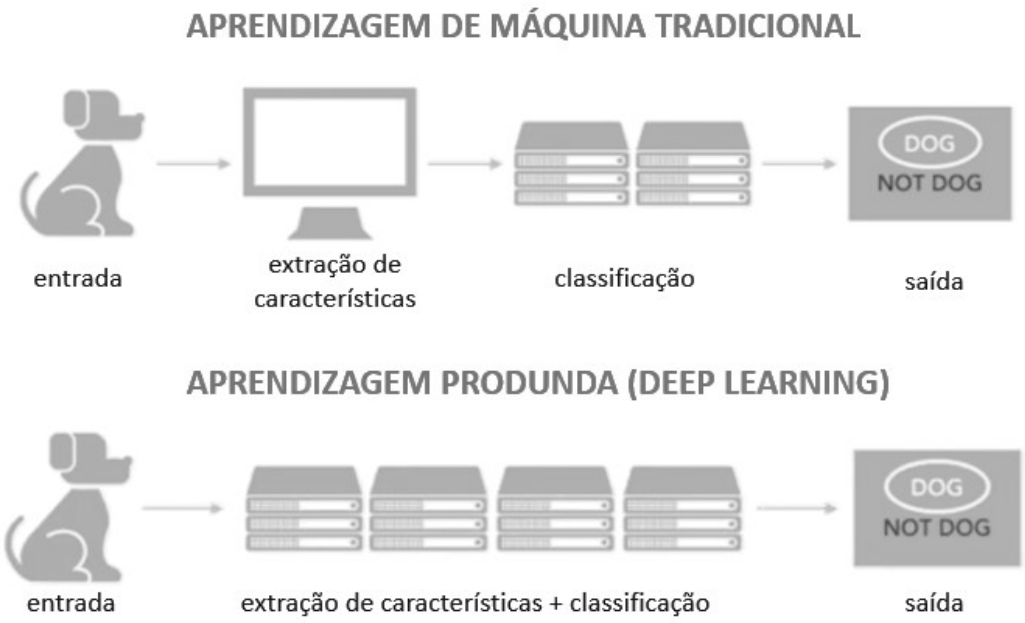

Figura 29 – Aprendizagem Profunda

2. *Deep Learning*, ou aprendizagem profunda, refere-se a uma classe de algoritmos que, baseada em modelos simplificados do funcionamento do cérebro, objetiva representar conceitos complexos em termos de conceitos mais simples, de forma hierárquica ou "multicamada". Cada "camada" de representação agrega conhecimento adquirido por camadas inferiores, que se especializam por sua vez em identificar os conceitos primitivos. Podemos imaginar, portanto, que quanto maior o número de camadas, ou melhor, quanto mais longa ou profunda for a sequência de camadas, maior a expressividade do modelo – daí a origem do termo *deep learning*.

3. Um exemplo desse tipo de entendimento hierárquico é a interpretação de uma imagem por um computador: as primeiras camadas de representação tendem a reconhecer padrões de pixels adjacentes – por exemplo, em termos de cores, para identificação de contornos. As camadas subsequentes tendem a especializar-se em reconhecer conceitos mais elaborados (ex.: vértices como

[1] O resumo apresentado nesta seção é baseado parcialmente em tradução livre do livro (Goodfellow, Bengio, & Courville, Deep Learning, 2016).

conjuntos de contornos) e posteriormente formas geométricas (constituídas por sua vez por padrões específicos de contornos e vértices). Já as próximas camadas tendem a reconhecer conceitos mais complexos formados a partir de determinados padrões de formas geométricas (ex.: diferenciar um pneu de um prato, uma vez que ambos têm formato circular), aprendendo a reconhecer padrões de textura, por exemplo. E assim, sucessivamente, por meio de múltiplas camadas de reconhecimento de padrões, o algoritmo chega ao nível de identificar objetos em cenas ou classificar imagens.

4. Para dados textuais, é possível pensar em hierarquias como sequências de caracteres que formam palavras de um vocabulário. Tais palavras por sua vez formam sentenças, que em seguida formam textos com determinados padrões semânticos que permitem classificar o assunto ao qual o texto se refere, ou então identificar, por exemplo, se um determinado texto é uma crítica ou um elogio, se contém ironia etc.

5. O exemplo mais simples de um modelo de *deep learning* é a rede neural *multilayer perceptron* (MLP), uma função matemática composta por inúmeras funções mais simples, que processam sequencialmente os dados de entrada – sejam eles um arquivo de imagem ou um documento textual - de modo que o valor computado por uma função é utilizado como valor de entrada pela função subsequente, e assim sucessivamente. Cada uma dessas funções é denominada de camada de representação. A figura abaixo apresenta um diagrama de alto nível que explica as camadas de entrada, ocultas e de saída em uma MLP:

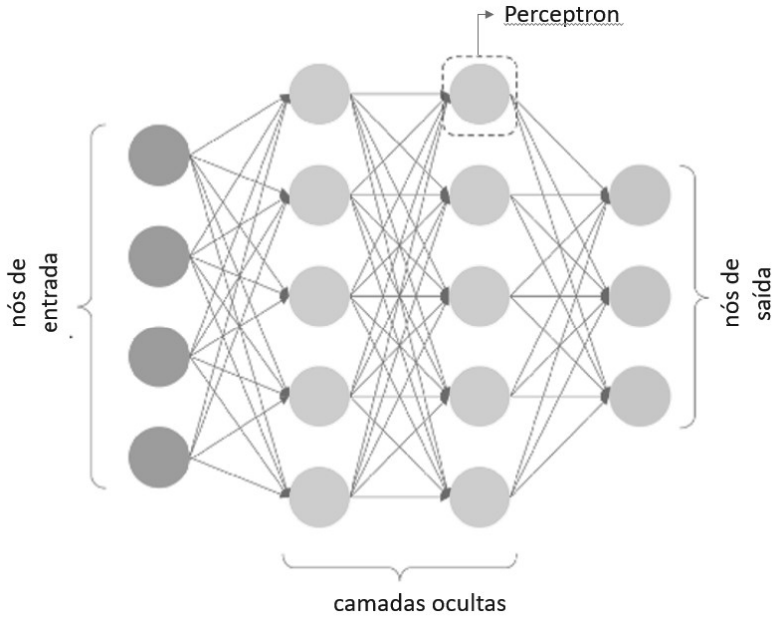

Figura 30 – Representação de uma rede neural *multilayer perceptron*

6. Por fim, é importante ressaltar que as redes neurais artificiais têm um histórico que remonta desde a década de 1940, com os primeiros modelos conceituais (McCulloch & Pitts, 1943), passando pelo algoritmo de retro propagação ou *backpropagation* (Rumelhart, Hinton, & Williams, 1986) - utilizado até os dias atuais – até chegar a meados do ano de 2006. Naquele ano, a terceira onda de pesquisa e uso de redes neurais foi impulsionada por inovações em termos de algoritmos e estratégias de treinamento, adaptáveis para diferentes arquiteturas de redes neurais e com maior capacidade não apenas de otimização, mas também de generalização.

7. O termo *deep learning* começa então a ser adotado, dada a importância teórica do maior número de camadas de unidades computacionais em redes neurais. Além disso, o maior poder computacional disponível na época veio a tornar viável o treinamento de redes neurais profundas. Um

exemplo é o advento do uso de unidades de processamento gráfico ou GPUs (*graphical processing units*), responsável por maior grau de paralelismo no processamento dos algoritmos de treinamento.

8. A inspiração biológica original – neurônios humanos – passou a ser cada vez mais substituída por inspiração advinda de outros campos, como álgebra linear, probabilidade, estatística, teoria da informação e otimização numérica. Além disso, o advento do Big Data, com um número cada vez maior de dispositivos computacionais produzindo grandes volumes de dados, ajudou a superar o maior obstáculo para a estatística estimativa: ser capaz de generalizar para novos dados depois de observar uma amostra. Assim, parte-se da premissa de que, aumentando a amostra, tem-se cada vez mais capacidade de aprendizado e generalização.

Tendências recentes em redes neurais

Transferência de Aprendizagem[1]

9. Transferência de aprendizagem (do inglês *transfer learning*) é uma técnica cada vez mais empregada, em especial nas áreas de visão computacional e processamento de linguagem natural, na qual o conhecimento adquirido por um modelo pré-treinado em um determinado domínio/conjunto de tarefas é "transferido" para outro domínio/conjunto de tarefas. Permite a "democratização" do uso de modelos de Inteligência Artificial, uma vez que novos modelos podem ser treinados com apenas uma fração da quantidade de dados e recursos computacionais que seriam utilizados caso um modelo tivesse que ser treinado "do zero".

10. A transferência de aprendizagem é ligeiramente inspirada na forma pela qual humanos aprendem, uma vez que dificilmente aprendemos algo do zero, mas comumente aprendemos por meio de analogia, incorporando a experiência adquirida anteriormente em problemas e situações similares a novos contextos.

Transferência de aprendizagem em processamento de linguagem natural (PLN)

11. Não restam dúvidas de que as arquiteturas e estratégias de treinamento de redes neurais adotadas nos últimos levaram a consideráveis avanços em tarefas como tradução de textos, respostas a perguntas (*question answering*) e *chatbots*, mesmo em tarefas treinadas do zero. Entretanto, mudanças significativas na distribuição amostral dos dados levavam à degradação de desempenho, indicando que os modelos haviam se especializado em desempenhar bem apenas com entradas de um determinado tipo (ex.: idiomas ou tipos de texto específicos).

12. Restavam desafios a serem superados para idiomas menos populares em comparação com o inglês, ou mesmo tarefas mais específicas ou ainda não exploradas. No caso concreto de idiomas, atente-se para o problema de línguas pouco faladas que, como consequência, contam com reduzida disponibilidade de corpus rotulado para treinamento de modelos de PLN.

13. Na década de 1960, um primeiro passo rumo à transferência de aprendizagem consistiu no uso de espaços vetoriais para representar palavras como vetores de números.

14. A metade da década de 2010 assistiu ao advento de modelos como word2vec (Mikolov, Chen, Corrado, & Dean, 2013), (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), sent2vec (Pagliardini, Gupta, & Jaggi, 2018) (Gupta, Pagliardini, & Jaggi, 2019) e doc2vec (Le & Mikolov, 2014). Tais modelos eram treinados para representar, respectivamente, palavras, sentenças e documentos em espaços vetoriais de forma que a distância entre os vetores estivesse relacionada à diferença de significado entre as entidades correspondentes. O treinamento para que essa vetorização fosse obtida

[1] O resumo apresentado nesta seção é parcialmente baseado em (Azunre, 2021)

utilizava como premissa atrelar o significado de uma palavra ao seu contexto, ou seja, às palavras adjacentes no texto. Tem-se aqui portanto um exemplo de treinamento não supervisionado.

15. Uma vez que as palavras, sentenças ou parágrafos estejam representados como vetores, é possível utilizar, por exemplo, algoritmos de classificação ou *clustering*, para os quais os dados de entrada são representados tão somente como pontos em um espaço vetorial. No exemplo concreto de classificação, tem-se, portanto, uma abordagem semisupervisionada, uma vez que, apesar da tarefa de classificação ser supervisionada, a representação dos dados de entrada foi obtida de forma não supervisionada, sendo ainda assim capaz de embutir a semântica textual.

16. Posteriormente, para lidar com palavras não previstas no vocabulário inicial (ex.: palavras novas, gírias, *emojis*, estrangeirismos ou nomes de pessoas), passou-se a utilizar vetorização a nível de caracteres.

17. A descrição acima pode ser entendida como uma forma embrionária de transferência de aprendizagem, uma vez que o modelo pré-treinado de vetorização já é capaz de embutir um certo nível de semântica ou significado a palavras, sentenças etc.

18. O ano de 2018 assistiu a uma verdadeira revolução na área de PLN, quando pesquisadores começaram a empregar transferência de aprendizagem em um nível mais abstrato, disponibilizando não mais modelos de vetorização pré-treinados, porém redes neurais inteiras pré-treinadas em tarefas genéricas, não supervisionadas, e ainda assim de mais alto nível. Como exemplo, temos redes neurais que implementam modelos de linguagem, que nada mais são do que modelos estatísticos treinados para prever a próxima palavra ou conjunto de palavras dados os termos anteriores. Em um processo conhecido como fine-tuning, é possível, obtendo um desses modelos pré-treinados, apenas fazer breves treinamentos adicionais, voltados para otimizar o modelo para a tarefa específica que se deseja treinar, ajustando os pesos da rede. A esse movimento dá-se inclusive o nome do "momento ImageNet", em referência ao uso até então bastante difundido de redes neurais pré-treinadas na base de imagens ImageNet (Stanford Vision Lab, Stanford University, Princeton University, 2021) para aplicações das mais diversas em visão computacional.

19. A tabela a seguir mostra um resumo das principais inovações consideradas pioneiras na área de transferência de aprendizagem para PLN, e que são em sua maioria utilizadas atualmente. Como o objetivo aqui é tão somente motivar e fornecer uma visão geral do estado da arte na área, o detalhamento da arquitetura, princípios matemáticos e algoritmos de treinamento de cada um desses modelos está fora do escopo deste documento e pode ser consultado nas referências:

Embeddings from Language Models (ELMo) (Peters, et al., 2018)	Um problema da formulação original do word2vec era a desambiguação: um único vetor representava cada palavra de um vocabulário, mesmo que tal palavra pudesse ter diferentes significados a depender do contexto. O ELMo resolveu esse problema gerando vetores contextualizados, a partir do treinamento de um modelo de linguagem. É possível utilizar grandes bases de texto não supervisionadas como por exemplo Wikipedia para o treinamento do modelo de linguagem.
Universal Language Model Fine-tunning (ULMFit) (Howard & Ruder, 2018)	Modelo voltado para adaptar qualquer modelo de linguagem baseado em rede neural para qualquer tarefa específica, tendo sido incialmente demonstrado em tarefas de classificação textual. Utiliza como estratégia para *fine-tuning* o uso de

	diferentes taxas de aprendizagem para as diferentes camadas da rede.
OpenAI Generative Pretrained Transforer (GPT) (Radford & Narasimhan, Improving Language Understanding by Generative Pre-Training, 2018)	Baseado no modelo de rede neural denominado *Transformer* (Vaswani, et al., 2017), grosso modo uma arquitetura de rede que permite maior paralelismo e desempenho em relação a arquiteturas anteriores, que além de não suportarem o mesmo grau de paralelismo, apresentavam dificuldade para lidar com textos longos. Em sua formulação mais recente – o GPT-3 (Brown, et al., 2020) – é capaz de gerar automaticamente textos realistas, semelhantes aos que seriam escritos por humanos, além de suportar recursos como aprendizado multitarefa, *one-shot learning*[1] e *zero-shot learning*[2].
Bidirecional Endoder Representations from Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2018)	Baseado no mesmo modelo de rede neural do GPT, porém com variações. Utiliza mascaramento de palavras a serem inferidas, adotando a acurácia dessa predição como métrica para o treinamento.

20. Para todos os modelos acima, foi demonstrado que era possível adaptá-los para tarefas mais especializadas, porém com reduzida quantidade de dados rotulados, não requerendo, portanto, alto custo em termos de base de textos para treinamento. Atualmente, muitos modelos de linguagem pré-treinados em vários idiomas estão disponíveis em repositórios, de forma aberta, para que possam ser livremente retreinados em tarefas específicas. Um exemplo desse tipo de repositório é o disponibilizado pela biblioteca HuggingFace (Huging Face, 2021), que disponibiliza milhares de modelos inclusive para tarefas especializadas como sumarização, classificação, geração de textos, tradução etc.

Generative Adversarial Networks

21. *Generative Adversarial Networks*, também conhecidas pelo acrônimo GANs ou, em tradução livre, redes generativas adversariais, referem-se a uma classe de arquitetura de redes neurais introduzida por um artigo de Ian Goodfellow e demais pesquisadores em 2014 (Goodfellow, et al., 2014).

22. Como o próprio nome dá a entender, modelos generativos especializam-se na geração de dados, sejam eles imagens, vídeos, textos, músicas ou mesmo discursos falados, de forma a assemelhar-se a dados criados por humanos, diferentemente de modelos discriminativos, que se especializam em classificar ou "compreender" informações (ex.: classificação de imagens ou detecção de objetos em aplicações de visão computacional, compreensão de textos em processamento de linguagem natural etc.).

23. Exemplos de aplicações de redes GANs têm sido noticiados pela mídia nos últimos anos, como por exemplo algoritmos capazes de gerar fotos de pessoas que não existem, ou fotos de ambientes (paisagens, cenários internos etc.) igualmente inexistentes (Karras, et al., 2020). Um exemplo de uso desvirtuado desse tipo de tecnologia inclui deep fake – uma prática que pode resultar, por exemplo, na criação de fotos para perfis falsos de redes sociais ("robôs"), montagens e manipulação de imagens fotográficas ou audiovisuais, geração de falsos discursos atribuídos a pessoas públicas etc. Por outro lado, bons usos de redes GANs incluem, por exemplo, a geração de obras de arte por artistas-programadores, e até mesmo usos na área de medicina, como a geração de imagens sintéticas para

[1] Técnica de treinamento na qual apenas um ponto de dado rotulado é necessário. Entradas – descritas em linguagem natural - são incluídas para instruir o modelo sobre qual tarefa deverá ser desempenhada.

[2] Técnica de treinamento na qual nenhum dado rotulado é necessário.

aumentar bases de treinamento para algoritmos de diagnóstico a partir de imagens (Frid-Adar, Klang, Amitai, Goldberger, & Greenspan, 2018).

APÊNDICE III – Principais domínios de aplicação de IA

Predição

0. Um Modelo Preditivo é, de forma simplificada, uma função matemática que pode ser aplicada a uma grande quantidade de dados soltos, para que seja possível a identificação de padrões que possam mostrar tendências futuras. É como se fosse possível prever com eficiência o futuro, de forma matemática, com probabilidade, estatística etc.

1. O Modelo Preditivo usa dados, algoritmos estatísticos e técnicas de Aprendizagem de Máquina para identificar a probabilidade de resultados futuros, a partir de dados armazenados em um determinado histórico.

2. Ter um Modelo Preditivo bem-feito e calibrado faz com que riscos e oportunidades sejam identificados com antecedência suficiente para a tomada de decisão mais adequada. Os exemplos de uso são muitos, e vão desde a prevenção contra fraudes e otimização de campanhas de marketing até melhorias em processos operacionais.

3. O Modelo Preditivo, em última análise, serve para embasar decisões, que se tornam mais eficientes por serem realizadas de acordo com um cenário de necessidades específicas, já que o modelo é moldado às necessidades de quem o criou e o alimenta.

4. Para criar um modelo preditivo, algumas etapas precisam ser cumpridas. O Apêndice IV – Ciclo de desenvolvimento de soluções de IA aborda esse tema.

Visão Computacional

5. Visão computacional é um domínio de aplicação e área de estudos que precede o aprendizado de máquina e envolve como os computadores enxergam, entendem e extraem informações relevantes sobre imagens e vídeos. A visão computacional tenta mimetizar a habilidade biológica humana de enxergar e compreender o ambiente ao seu redor para que possa tomar decisões baseadas nesses dados.

6. A visão computacional tem aplicação útil na Administração Pública em áreas que necessitem de análises visuais como: acompanhamento de obras por imagens de satélite como rodovias, usinas e qualquer outro tipo de objeto que possa ser capturado por satélites com resoluções cada vez mais altas; reconhecimento de áreas e busca de sobreviventes com drones por meio do reconhecimento de pessoas pelos vídeos capturados; reconhecimento facial de pessoas procuradas; reconhecimento de placas de veículos, detecção precoce de doenças como diabetes, alguns tipos de câncer, Alzheimer, entre vários outros casos na área médica a partir de análise de imagens.

7. A área de visão em seu período anterior ao uso de aprendizado de máquina se baseava em algoritmos determinísticos com filtros para detecção de bordas, limites, curvas, cantos, entre outros para que fosse possível compor uma representação e entendimento de mais alto nível. Em 1989, Yann LeCun construiu uma solução para o problema baseada em uma nova tecnologia à época denominada redes convolucionais, para reconhecimento de códigos de endereçamento postal (CEPs) escritos a mão em cartas nos Estados Unidos, com o objetivo de automação desta atividade. Abaixo está representada uma rede Convolucional para reconhecimento de dígitos de 0 a 9.

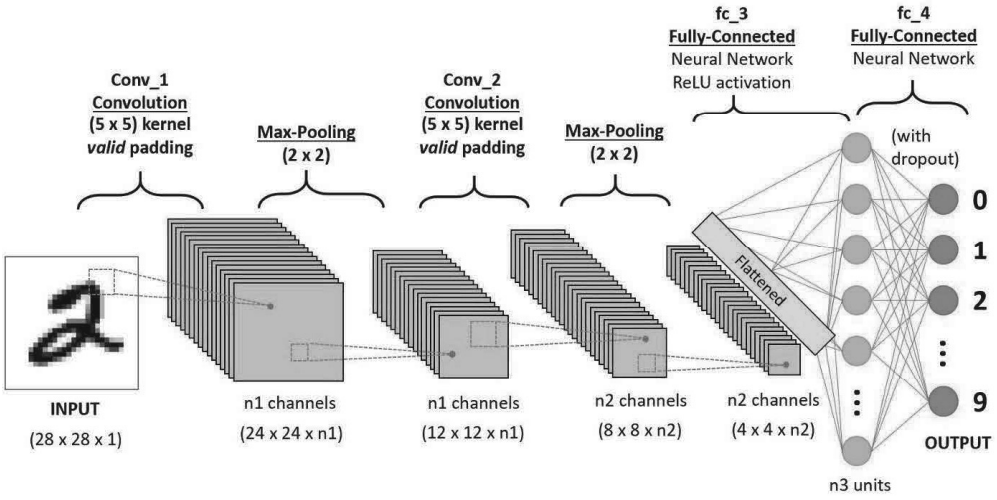

Figura 31 – Representação do reconhecimento de números.

8. A grande popularização recente da área de visão computacional aconteceu a partir de 2012, com o advento de competições como a *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) (https://www.image-net.org/challenges/LSVRC/), que buscam a avaliação de soluções nesta tecnologia.

9. Nos anos que se seguiram, a competição foi dominada por modelos de aprendizado profundo, baseando-se principalmente em arquiteturas de redes convolucionais cada vez mais profundas. As taxas de erro de classificação foram baixando a ponto de superar o nível humano médio. Com os resultados melhores e as redes cada vez mais profundas e demoradas para se treinar, a transferência de aprendizagem foi ganhando cada vez mais espaço para reaproveitar o treino prévio e apenas refinar os pesos da rede em novas aplicações sem a necessidade de treinar do zero.

10. Mesmo com os significativos avanços dos últimos anos em detecção de objetos em imagens, o problema está muito longe de ser considerado resolvido quando comparado com o que é feito por seres humanos. Exemplos disso incluem a descrição de imagens com inúmeros objetos interagindo entre si, entendimento de vídeos ou sequências de vídeos longos como filmes e séries contextualizando o que é visto com o que é de conhecimento geral de pessoas para generalizar suas observações.

11. Uma compilação de artigos publicados na área de visão computacional com o estado da arte e código fonte da implementação está disponível em https://paperswithcode.com/area/computer-vision.

Processamento de Linguagem Natural

12. Processamento de linguagem natural (PLN) é uma subárea da ciência de computação, mais especificamente, da inteligência artificial, que está associada aos problemas não apenas de processamento, mas também de compreensão da linguagem natural humana em seus vários idiomas, e de geração de linguagem. Engloba tanto a linguagem falada como a escrita. Suas aplicações envolvem desde a análise e processamento de texto escrito até processamento e geração de diálogos entre humanos e agentes não humanos.

13. O uso de PLN é uma área de forte aplicabilidade no serviço público, dada principalmente a importância da linguagem escrita (normativos, documentos digitalizados etc.), em que se vislumbra por exemplo o processamento automático de centenas ou milhares de documentos (ex.: peças de processos judiciais ou de órgãos de controle) em que a aplicação tão somente da capacidade humana tem-se mostrado inviável, o que historicamente tem contribuído para a tão conhecida lentidão na tramitação de processos e outros assuntos diretamente relacionados à vida dos cidadãos.

14. Os primeiros usos do que hoje vem a ser conhecido como processamento de linguagem natural fazia uso quase exclusivo de reconhecimento determinístico de padrões por meio de expressões regulares, um tipo de programação baseado portanto em regras bem definidas. Posteriormente, os usos de aprendizagem de máquina foram focados no treinamento de modelos tradicionais como regressão logística, Naïve Bayes[1], máquinas de vetores de suporte, entre outros, em que os dados de entrada continham inúmeras combinações de atributos que visavam a caracterizar completamente os textos (ex.: classificação gramatical de cada palavra obtida por algoritmos específicos, classificação gramatical das *N* palavras subsequentes e/ou anteriores etc.), em um processo conhecido como engenharia de atributos (*feature engineering)*. A escolha das características ou *features* a serem utilizadas era um processo manual, de alto custo, que consumia a maior parte do tempo alocado ao desenvolvimento de modelos de aprendizagem. Nos últimos anos, redes neurais têm sido empregadas largamente não somente na etapa de aprendizado de tarefas propriamente dito, em substituição aos algoritmos citados no parágrafo anterior, como principalmente na geração da representação do texto de entrada. Em suma, o desenvolvedor não mais se preocupa em determinar as características de cada palavra do texto a ser analisado, mas delega à rede a tarefa de, dada uma sequência de palavras ou subpalavras presentes em um vocabulário, gerar a representação mais adequada à semântica do texto ou discurso falado, para que essa representação seja utilizada pela mesma rede ou até mesmo por outro algoritmo para a tarefa desejada (ex.: classificação textual).

15. Por fim, (Jurafsky & Martin, 2020) chama atenção para possíveis maus usos de modelos de inteligência artificial, a exemplo do modelo GPT-3 (Brown, et al., 2020), uma rede neural geradora de textos criada pela OpenAI. Considerado por alguns como "a inteligência artificial mais avançada já criada pelo homem" (Rodrigues, 2021), o GPT-3 tem levantado sérios questionamentos éticos devido à incrível semelhança dos textos por ela gerados em relação a textos escritos por humanos.

Principais aplicações e técnicas de processamento de linguagem natural (PLN)

1. Aplicações supervisionadas

1.1. Classificação textual e análise de sentimentos

16. Uma classe de tarefas facilmente compreensível e comum em inúmeros cenários de uso de processamento de linguagem natural certamente inclui classificação de documentos, definida em (Ruder, Text classification, 2019) como "a tarefa responsável por atribuir a uma sentença ou documento uma categoria apropriada, dentre uma lista de categorias dependente da base de dados" (tradução livre) ou da aplicação específica.

17. Modelos baseline para algoritmos de classificação textual em geral incluem Naïve Bayes, regressão logística e máquinas de vetores de suporte. Tais modelos costumam ser a primeira escolha e em muitas situações são de fato os mais adequados, não apenas pelo desempenho em termos computacionais, mas também por atingirem acurácia suficiente para o cenário pretendido de uso, além de não exigirem recursos como GPU ou computação em nuvem.

18. Em (Ruder, Text classification, 2019), é disponibilizada uma lista de bases de dados para treinamento de modelos para a língua inglesa, bem como respectivos modelos estado-da-arte. Dentre tais modelos, vemos que no momento da escrita deste relatório, é comum o uso de tecnologia de rede neural baseada em arquiteturas como *Transformers* (Yang, et al., 2019), ao lado de redes recorrentes.

19. Uma tarefa relacionada à classificação textual, que pode ser vista como uma especialização, é a chamada **análise de sentimentos**, um tipo de aplicação em geral voltado para analisar opinião de

[1] Naive Bayes classifier: https://en.wikipedia.org/wiki/Naive_Bayes_classifier

usuários e verificar a polaridade de opiniões ou sentimentos. Aqui, aplicam-se basicamente os mesmos algoritmos empregados em classificação textual, tanto para algoritmos de baseline como estado-da-arte.

20. Como exemplo de *dataset* para o idioma português, temos o TweetSentBR Dataset (Brum & das Graças Volpe Nunes, 2018), além de um corpus de avaliação de hotéis (Freitas & Vieira, 2015).

1.2. *Part of speech recognition*

21. Reconhecimento de partes de um discurso (ou **part of speech recognition**) refere-se à tarefa de classificar cada termo ou palavra em um texto em uma determinada categoria gramatical (ex.: adjetivo, advérbio, substantivo comum/próprio, preposição, interjeição, verbo auxiliar, numeral, pronome, sinal de pontuação etc.). Segundo (Jurafsky & Martin, 2020), ao lado do reconhecimento de entidades mencionadas, é uma tarefa capaz de fornecer pistas úteis para indicar a estrutura e significado de sentenças.

22. O reconhecimento de partes de um discurso – referenciado deste ponto em diante pelo acrônimo inglês *POS tagging* – pertence a uma classe de tarefas denominada rotulagem de sequências. São tarefas que atribuem a cada palavra x_i em uma sequência de entrada X um rótulo y_i (ex.: verbo ou substantivo etc.), gerando, portanto, uma nova sequência Y com o mesmo tamanho da sequência de entrada X. As soluções utilizadas para este fim incluem desde algoritmos probabilísticos/estatísticos clássicos como **Hidden Markov Model**[1] com algoritmo **Viterbi**[2] e **Conditional Random Field**[3] até algoritmos modernos baseados em redes neurais recorrentes (Wang, et al., 2021) ou, mais recentemente, redes baseadas em *Transformers* (Heinzerling & Strube, 2019) ou até mesmo combinações inteligentes de *embeddings* (Wang, et al., 2021).

1.3. Reconhecimento de entidades mencionadas

23. **Reconhecimento de entidades mencionadas**, tradução livre para *named entity recognition* (**NER**) é a tarefa de rotular entidades em um texto com seu tipo correspondente. Aqui, "tipo de entidade" não significa classe gramatical ou sintática, mas a categorização de entidades que em geral referem-se a nomes próprios (podendo, entretanto, referenciar conceitos mais abstratos). Os tipos frequentemente encontrados em modelos que classificam entidades mencionadas incluem pessoa, lugar, data etc.

24. Tradicionalmente, modelos de baseline para reconhecimento de entidades mencionadas faziam uso de algoritmos estatísticos como Conditional Random Field, muitas vezes em combinação com redes neurais recorrentes LSTM ou convolucionais.

25. No momento da escrita deste relatório, soluções consideradas estado-da-arte utilizavam *Transformers* (BERT) (Yamada, Asai, Shindo, Takeda, & Matsumoto, 2020) e combinação de rede neural recorrente do tipo LSTM bidirecional com rede convolucional (Shahzad, Amin, Esteves, & Ngonga Ngomo, 2021).

26. Em termos de *dataset* para reconhecimento de entidades na língua portuguesa, temos o HAREM (Mota & Santos, 2008).

1.4. *Chatbots*

27. Em (Jurafsky & Martin, 2020), **chatbots** são definidos como "sistemas projetados para conversações estendidas, de forma a mimetizar conversações não estruturadas características de

[1] Hidden Markov Model: https://en.wikipedia.org/wiki/Hidden_Markov_model

[2] Viterbi algorithm: https://en.wikipedia.org/wiki/Viterbi_algorithm

[3] Conditional Random Field: https://en.wikipedia.org/wiki/Conditional_random_field

interações entre humanos, principalmente para fins de entretenimento, mas também para propósitos práticos como tornar agentes orientados a tarefas mais naturais" (tradução livre).

28. Ainda segundo (Jurafsky & Martin, 2020), as três principais arquiteturas para chatbots incluem sistemas baseados em regras, sistemas de recuperação de informações e geradores *encoder-decoder*.

29. **Sistemas baseados em regras** podem ser definidos como aqueles que utilizam padrões textuais semelhantes a expressões regulares para interpretar as perguntas feitas pelo usuário e gerar as respostas mais adequadas. ELIZA (ELIZA, 2021) e A.L.I.C.E (A.L.IC.E, 2019) são exemplos históricos desse tipo de chatbot. Apesar da simplicidade desses sistemas, uma vez que em geral não empregam nenhum mecanismo de aprendizagem de máquina propriamente dito, até hoje alguns chatbots ainda utilizam essa arquitetura básica, também denominada arquitetura "padrão/ação" (Jurafsky & Martin, 2020), sendo alguns capazes até mesmo de serem aprovados no Teste de Turing[1].

30. **Sistemas de recuperação de informações** se assemelham a motores de busca, pois tratam a pergunta realizada pelo usuário como uma *query*. Tais sistemas em geral utilizam técnicas baseadas em similaridade textual.

31. **Geradores encoder-decoder** são uma tentativa de tratar o problema de geração de diálogos por meio de arquiteturas do tipo *encoder-decoder*[2]. Modelos alternativos incluem o *fine-tuning* de um modelo de linguagem em um *dataset* de diálogos, a exemplo do que foi feito em (Paranjape, et al., 2020), com *fine-tuning* do GPT-2 (Radford, et al., 2019). Segundo (Jurafsky & Martin, 2020), essa classe de solução tem foco na geração de respostas únicas, não sendo apropriadas para um fluxo de diálogo longo, em que teriam que ser aplicadas técnicas adicionais baseadas por exemplo em aprendizagem por reforço ou redes adversariais.

32. A classificação acima está longe de ser exaustiva, uma vez que cada vez mais as soluções para construção de *chatbots* tendem a empregar arquiteturas hibridas que empregam desde sistemas baseados em regras e expressões regulares em combinação com aprendizagem de máquina com uso de dados até técnicas de aprendizagem por reforço para prever a próxima pergunta a ser feita ao usuário ou próxima ação a ser executada (ex.: reserva de voos, hotéis etc.).

33. No momento da escrita deste relatório, trabalhos considerados estado-da-arte incluíam o uso de redes neurais recorrentes (Li, Lin, Collinson, Li, & Chen, 2019) e *Tranformers*/BERT (Lai, Hung Tran, Bui, & Kihara, 2020) (Liu, et al., 2020).

1.5. Outras aplicações supervisionadas

34. A tabela abaixo lista outros tipos de aplicações supervisionadas em processamento de linguagem natural que não foram detalhadas neste relatório, mas que sem dúvida merecem atenção tanto em termos de aplicabilidade como para exploração dos limites do uso de inteligência artificial para lidar com as complexidades inerentes à linguagem humana. Os exemplos com as respectivas descrições foram baseados em (Ruder, NLP-progress, 2021), bem como as referências[3] para as implementações

[1] Dizemos eu um sistema é aprovado no Teste de Turing quando um usuário, ao travar uma conversação com tal sistema, não é capaz de diferenciá-lo de um ser humano.

[2] Nesse tipo de arquitetura neural, resumidamente, uma rede codificadora é treinada para gerar uma representação vetorial intermediária do texto de entrada, enquanto uma rede decodificadora é treinada para resolver a tarefa-alvo (ex.: tradução de textos ou, no caso de chatbots, geração de respostas) a partir da representação intermediária.

[3] Dentre as referências aqui citadas, nem todas foram publicadas oficialmente em conferências ou periódicos, estando algumas em fase "*preprint*". Ainda assim, o objetivo aqui não é entrar no mérito de cada trabalho, mas sim apontar os progressos mais recentes na área, que tem evoluído constantemente.

estado-da-arte segundo o site. Para a listagem completa de tarefas em PLN com respectivas referências, consultar (Ruder, NLP-progress, 2021)

Tarefa	Descrição	Principais tipos de tecnologias empregados nos últimos dois anos	Referências para exemplos de trabalhos estado-da-arte
Extração de relacionamentos	Extração de relacionamentos semânticos de um texto. Relacionamentos extraídos normalmente ocorrem entre duas ou mais entidades de um certo tipo (ex.: pessoa, organização, local) e recaem em um número de categorias semânticas (ex.: "casado com", "empregado por", "mora em").	*Transformer,* *LSTM*	(Baldini Soares, FitzGerald, Ling, & Kwiatkowski, 2019) (Yamada, Asai, Shindo, Takeda, & Matsumoto, 2020) (Nayak & Ng, 2020)
Geração de paráfrase	Geração de uma sentença de saída que preserva o significado da sentença de entrada, mas contém variações em escolha de palavras e gramática.	*Transformer*	(Bui, Le, To, & Cha, 2021)
Question Answering	Resposta a perguntas a partir da compreensão de um texto escrito.	*Transformer*	(Yang, et al., 2019)
Reconhecimento automático de discurso	Compreensão e/ou transcrição de discurso falado	Redes neurais convolucionais *Transformer* Redes neurais recorrentes	(Hsu, et al., 2021) (Gulati, et al., 2020)
Resolução de correferência	Agrupamento de menções no texto que se referem às mesmas entidades do mundo real	*Transformer*	(Kirstain, Ram, & Levy, 2021) (Attree, 2019)
Senso comum	Tarefas de raciocínio de senso comum requerem que o modelo vá além do reconhecimento de padrões, utilizando ao invés "senso comum" ou conhecimento do mundo para inferir suas predições. Exemplos incluem compreensão de textos, geração de textos que descrevam o que uma imagem dá a entender, entre outros.	*Transformer*	(Yang, et al., 2019)
Similaridade textual semântica	Determina o grau de similaridade entre dois textos.	*Transformer*	(Yang, et al., 2019)
Sumarização	Produção de uma versão menor de um ou vários documentos,	*Transformer,* *Pointer-Generator network,*	(Zhong, et al., 2020) (Dou, Liu, Hayashi, Jiang, & Neubig, 2021)

	preservando a maior parte do significado original.	LSTM	
Tradução	Tarefa de traduzir uma sentença em uma linguagem de origem para uma linguagem de destino diferente.	*Transformer*	(Liu, Duh, Liu, & Gao, 2020) (Liu, Liu, Gao, Chen, & Han, 2020)
Visual Question Answering	Dada uma imagem e uma pergunta em linguagem natural, a tarefa consiste em fornecer uma resposta correta em linguagem natural.	*Transformer*	(Hu, Singh, Darrell, & Rohrbach, 2020)

2. Aplicações não supervisionadas

2.1. Agrupamento (*Clustering*)

35. De acordo com (Manning, Raghavan, & Schütze, 2009), *clustering* é a forma mais comum de aprendizado não supervisionado.

36. Embora os algoritmos para clustering citados no Apêndice I tenham sido originalmente concebidos tendo em mente técnicas de representação textual convencionais como TF-IDF[1], estratégias de vetorização mais recentes baseadas em BERT/*Transformers* começam a ser empregadas para representar sentenças/documentos para fins de agrupamento (Reimers, 2021).

1.1.1. Modelagem de tópicos

37. Modelagem de tópicos é outra técnica de aprendizado não supervisionado, também utilizada para agrupar documentos por similaridade. Sua principal aplicação prática consiste em agrupar documentos por "tópico" ou assunto/tema de interesse. Outra aplicação prática de alguns dos algoritmos aqui utilizados inclui sistemas de recomendação (ex.: recomendação de livros, filmes ou músicas/artistas que podem agradar a determinado usuário com base em seu histórico de atividades – sejam elas filmes assistidos, livros comprados etc).

38. Tradicionalmente, as tarefas de modelagem de tópicos têm sido resolvidas com métodos matemáticos especialmente no campo da álgebra linear, coletivamente referidos como métodos de decomposição ou fatoração de matrizes. Os principais exemplos incluem **Latent Semantic Analysis[2]**, baseada em uma técnica conhecida como **Singular Value Decomposition[3]**, e **Non-negative Matrix Factorization[4]**. Outra técnica empregada – **Latent Dirichlet Allocation[5]** – baseia-se por sua vez em algoritmos estatísticos generativos que objetivam modelar cada documento como uma mistura de tópicos (cada tópico em maior ou menor nível) e cada palavra por sua vez como pertencente a um desses tópicos. Detalhes matemáticos a respeito dessas técnicas podem ser encontrados em (Manning, Raghavan, & Schütze, 2009) e/ou nas respectivas referências.

39. Os algoritmos supracitados são antigos e em sua maioria baseados em vetorizações do *tipo bag of words*, em que cada documento é representado por um vetor em que cada posição está associada a uma palavra do vocabulário. Assim, cada posição do vetor contém o número de vezes em que a respectiva palavra ocorre no documento. Porém, nos últimos anos, técnicas de vetorização com uso de

[1] Tf-idf: https://pt.wikipedia.org/wiki/Tf%E2%80%93idf

[2] Latent semantic analysis: https://en.wikipedia.org/wiki/Latent_semantic_analysis

[3] Singular value decomposition: https://en.wikipedia.org/wiki/Singular_value_decomposition

[4] Non-negative matrix factorization: https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

[5] Latent Dirichlet allocation: https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

arquiteturas de rede neural baseadas em *Transformers* têm se tornado cada vez mais comuns, e começam a ser utilizadas também para representar documentos para fins de modelagem de tópicos, ainda que por vezes os mesmos algoritmos acima ainda sejam utilizados para a finalidade principal de agrupamento. Exemplos incluem BERTopic (Grootendorst, 2020), baseado por padrão em vetorização com BERT (embora suporte outros modelos) e com a vantagem de inferir o número ideal de tópicos, e Contextualized Topic Models (Bianchi, Terragni, & Hovy, Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence, 2021) (Bianchi, Terragni, Hovy, Nozza, & Fersini, Cross-lingual Contextualized Topic Models with Zero-shot Learning, 2021), que suporta uma combinação de *transformers* com *bag of words*, requerendo entretanto que se defina a priori a quantidade de tópicos. Por fim, Top2Vec (Angelov, 2020), para modelagem de tópicos e busca semântica, é outro exemplo de ferramenta recente que suporta, entre outras técnicas de vetorização, o uso de vetorização com BERT aliada a HDBSCAN para geração dos agrupamentos.

3. Outras bases de dados para a língua portuguesa

40. A maior parte da pesquisa em processamento de linguagem natural tem tido como foco *datasets* (bases de textos para treinamento de modelos) em idiomas mais comumente falados como inglês e mandarim, e algumas vezes espanhol, francês ou alemão. Por outro lado, o idioma português ainda tem uma carência de bases rotuladas para as várias tarefas disponíveis. Muitas vezes não é possível resolver tarefas para as quais já existe solução (algoritmos mais recentes baseados em inteligência artificial) devido à escassez de uma base de textos que tenha sido rotulada adequadamente.

41. A tarefa de rotulagem manual também é de alto custo, trabalhosa, passível de erros e de subjetividade, e por isso comumente não é uma opção para grande parte dos pesquisadores da academia e da indústria. Por outro lado, modelos de inteligência artificial ainda dependem fortemente de dados para atingirem melhor desempenho, uma vez que o status atual de maior parte do desenvolvimento em inteligência artificial ainda é baseado em aprendizagem de máquina (*machine learning*) supervisionada por dados.

42. Iniciativas já foram criadas para mitigar o problema de linguagens denominadas *low resource* (de baixos recursos), como transferência de aprendizagem (ver Apêndice II – Redes Neurais Artificiais - seção 0).

43. A seguir, apresentamos uma lista de outras iniciativas e *datasets* para tarefas diversas em PLN para a língua portuguesa:

a) *List of Portuguese Datasets for Machine Learning Projects*: https://metatext.io/datasets-list/portuguese-language

b) *PortugueseGLUE Dataset*: tradução para o português do *benchmark* GLUE (General Language Understanding Evaluation (Wang, et al., 2018), que consiste em dados para treinamento de nove tarefas de compreensão de linguagem): https://metatext.io/datasets/portugueseglue

c) PLN - PUCRS – Grupo de Processamento da Linguagem Natural – RECURSOS E FERRAMENTAS: https://www.inf.pucrs.br/linatural/wordpress/recursos-e-ferramentas/

d) Tradução: exemplos de *datasets* incluem o CAPES Dataset (Soares, Yamashita, & Anzanello, 2018) – um corpus paralelo de resumos (abstracts) de teses e dissertações em português e inglês da base da CAPES – além do projeto OPUS (Tiedemann, 2012).

e) Extração de relações: ReRelEM (Freitas, Santos, Mota, Gonçalo Oliveira, & Carvalho, 2009), Summ-it++ (Antonitsch, et al., 2016).

f) Resolução de correferência: Corref-PT (Fonseca, et al., 1998) e Summ-it++ (Antonitsch, et al., 2016).

g) BERTimbau, um modelo de linguagem pré-treinado com o algoritmo BERT (Devlin, Chang, Lee, & Toutanova, 2018) para a língua portuguesa (Souza, Nogueira, & Lotufo).

h) Tradução do SQuAD (The Stanford Question Answering Dataset[1]): SQuAD v1.1 Dataset (Portuguese SQuAD v1.1 Dataset, 2019).

i) Reconhecimento de discurso falado e síntese de fala: How2 (Sanabria, et al., 2018) contém uma coleção de vídeos instrucionais com legendas em inglês e traduções para português.

44. Paralelamente, vale ressaltar que o *Center for Artificial Intelligence – C4AI* – iniciativa sediada pela Universidade de São Paulo, em parceria com outras instituições acadêmicas e privadas – está conduzindo projetos com o objetivo de melhorar a disponibilidade de corpus para a língua portuguesa (C4AI, 2020).

[1] SQuAD: https://rajpurkar.github.io/SQuAD-explorer/