NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados

Master Program in Advanced Analytics

A criação de um modelo de Natural Language Processing para extração de habilidades técnicas na área de Ciência de Dados

Rennan Valadares Ornelas Araújo

Dissertação apresentada como requerimento parcial para obter o grau de mestre em Data Science e Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

A CRIAÇÃO DE UM MODELO DE NATURAL LANGUAGE PROCESSING PARA EXTRAÇÃO DE HABILIDADES TÉCNICAS NA ÁREA DE CIÊNCIA DE DADOS

por

Rennan Valadares Ornelas Araújo

Dissertação apresentada como requerimento parcial para obter o grau de mestre em Data Science e Advanced Analytics

Orientador: Mauro Castelli

Novembro 2021

DEDICATÓRIA

Devo inteiramente a Deus o sucesso deste trabalho. Sem Ele esta jornada não seria possível e por isto, lhe dedico esta pesquisa.

AGRADECIMENTOS

Ao meu orientador Mauro Castelli que me ajudou e guiou durante esta incrível jornada, dando me ensinamentos inestimáveis em suas aulas e me dando suporte para a conclusão desta dissertação.

A Nova IMS por ter dado todo o suporte necessário para o meu desenvolvimento e crescimento como aluno.

A minha esposa Lívia, por ser minha base e companheira de vida, sempre me incentivando, aconselhando e motivando.

A minha mãe Elody por ser meu exemplo e ter me ensinado a ser a pessoa que sou hoje.

Ao meu Pai por ter me dado a vida e me amado.

Aos meus irmãos e irmãs, que sempre me encorajaram a buscar os meus objetivos.

Ao Professor Gilmar Marques por ter me ajudado no percurso da minha formação em Administração.

RESUMO

O trabalho surgiu da necessidade do homem de suprir suas necessidades básicas. Na antiguidade, época de Gregos e Romanos, era ensinado a prole como cuidar da terra e esse conhecimento era passado por gerações. Com a chegada da primeira revolução industrial e posteriormente com a popularização dos computadores, o trabalho se tornou o que conhecemos hoje. A forma de contratação mudou ao longo dos anos e com a chegada da tecnologia mais do que nunca, o mercado de trabalho está em constante mudança e como consequência disto as habilidades necessárias para estes trabalhos também seguem esta mudança. Grande parte das vagas de trabalho hoje estão anunciadas em sites de buscas de emprego como *Indeed*, *Glassdoor* e *Linkedin*. Um grande desafio atual, é prever as mudanças e tendências do mercado de trabalho em termos de habilidades, e a analise textual pode gerar uma vantagem competitiva neste sentido.

A proposta deste trabalho é analisar através de técnicas de Natural Language Processing (NLP) diferentes oportunidades de emprego da área de Data Science a fim de obter um modelo que possa ser utilizado para extrair as habilidades requisitadas para esta área.

Para alcançar este objetivo primeiro é feita uma revisão dos conceitos de Aprendizado de Máquina, Natural Language Processing, Transfer Learning e das técnicas de preparação de dados, e em seguida será apresentada a metodologia utilizada. Depois, são destacadas as técnicas que funcionam melhor para extração de habilidades, a escolha e criação do modelo, e por fim a apresentação de resultados.

PALAVRAS-CHAVE

Aprendizagem de Máquina; Aprendizagem Profunda; Habilidades; Habilidades Técnicas; Inteligência Artificial; IA; Mercado de Trabalho; Mineração de Texto; PLN; Processamento de Linguagem Natural; Recursos Humanos.

ABSTRACT

The work originated from man's need to meet his basic needs. In ancient times, the times of the Greeks and Romans, offspring were taught how to take care of the land and this knowledge was passed on for generations. With the arrival of the first industrial revolution and later with the popularization of computers, work became what we know today. The way of hiring has changed over the years and with the arrival of technology more than ever, the job market is constantly changing, and consequently, the skills needed for these jobs also follow this change. A large part of job vacancies today is advertised on job search sites such as Indeed, Neuvoo, and Linkedin. A big challenge today is to predict the changes and trends in the job market in terms of skills and textual analysis can generate a competitive advantage in this sense.

The purpose of this work is to analyze through Natural Language Processing (NLP) techniques different job opportunities in the Data Science area to obtain a model that can be used to extract the required skills for this area.

To achieve this goal first a review of the concepts of Machine Learning, Natural Language Processing, Transfer Learning, and data preprocessing, then the methodology used is presented. Next, the techniques that work best for skill extraction is highlighted, the choice and creation of the model, and finally the presentation of results.

KEYWORDS

Machine Learning; Deep Learning; Skills; Technical Skills; Artificial Intelligence; AI; Job Market; Text Mining; NLP; Natural Language Processing; Labor.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

ÍNDICE DE EQUAÇÕES

LISTA DE SIGLAS E ABREVIATURAS

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
DL	Deep Learning
EMR	Electronic Medical Records
FN	False Negatives
FP	False Positives
GPT	Generative Pre-trained Transformer
GPU	Graphics Processing Unit
HTML	Hyper Text Markup Language
ICA	Independent Component Analysis
KNN	K Nearest Neighbors
KPCA	Kernel based Principal Component Analysis
LSTM	Long Short-Term Memory
ML	Machine Learning
MMDBM	Mixed Mode Data Base Miner
NER	Named Entity Recognition
NLP	Natural Language Processing
NLTK	Natural Language Tool Kit
PCA	Principal Component Analysis
QA	Question Answering
ROC	Receiver Operating Characteristic
SVM	Support Vector Machines
TF-IDF	Term Frequency - Inverse Document Frequency
TN	True Negatives
TP	True Positives
USPTO	United States Patent and Trademark Office

1. INTRODUÇÃO

Com o aumento do uso de tecnologia nos processos de contração de pessoas, o mercado de trabalho tem se tornado volátil. Em **(Kon. 2017)**, a autora cita uma quarta revolução industrial marcada pela introdução de novas tecnologias no mercado, que vem alterando de forma significativa a mão de obra disponível. Em **(Pinzone et al., 2017)** os autores deixam claro o interesse e a importância da implementação da Indústria 4.0 e citam novas tecnologias como a internet das coisas, computação em nuvem e Big Data como propulsores de uma mudança que já está acontecendo.

Uma forma de acompanhar esta volatilidade no mercado de trabalho é por meio da identificação de habilidades em vagas de emprego. Ao estabelecer quais são as habilidades usuais requeridas para uma área de trabalho é possível entender as mudanças ao longo do tempo.

A mineração de textos é uma ferramenta poderosa para auxiliar na identificação de habilidades. Em **(Zanini & Dhawan, 2015)** os autores descrevem a mineração de texto como sendo uma forma estatística e computacional de análise de textos e explicam que a mineração de texto engloba várias técnicas que podem ser combinadas para solução de um problema.

Natural Language Processing, ou NLP, é um componente da mineração de textos e de acordo com **(Liddy, 2001)**, NLP é uma técnica computacional que visa entender a linguagem escrita ou falada por seres humanos para uma gama de tarefas ou aplicações. O campo de NLP é extenso e tem muitas aplicações como classificação de textos, análise de sentimentos, tradução de idiomas, deteção de *spams* e reconhecimento de entidades nomeadas.

O objetivo deste projeto é a criação de um modelo de Natural Language Processing para extração de habilidades técnicas na área de ciência de dados. Seguindo o objetivo principal três objetivos específicos foram definidos:

- Entender quais as técnicas de pré processamento de dados melhor se enquadram para este problema;
- Explorar os modelos de aprendizado de máquina a fim de compreender as diferentes técnicas de extração de habilidades em vagas de emprego;
- Escolher e aplicar a técnica que se mostrar mais promissora.

2. REVISÃO DA LITERATURA

2.1. APRENDIZADO DE MÁQUINA

Nesse tópico serão abordadas as classes de algoritmos de aprendizagem de máquina voltadas para a resolução de tarefas. Aprendizagem supervisionada e não supervisionada tem aplicações bem distintas no seu objetivo e será possível discorrer sobre a importância de cada uma delas em seu contexto próprio.

2.1.1. Aprendizado Supervisionado

O conceito de aprendizado supervisionado, do inglês Supervised Learning, está vinculado a relação entre um conjunto de variáveis de entrada que gera uma variável de saída, e que gera um mecanismo que torna possível a predição de novos valores a partir de dados que não foram ainda vistos **(Cunningham et al., 2008)**, sendo este conjunto de técnicas considerado o mais importante de aprendizagem de máquina. Em **(Liu, 2011)** essa definição vai além ao indicar uma importante distinção dentro dos algoritmos de aprendizagem supervisionada relacionada ao tipo de saída dos modelos: um modelo com um conjunto finito de saídas discretas é chamado de um algoritmo de classificação, enquanto um modelo com saídas de valor contínuo é denominado como um algoritmo de regressão. Ainda neste trabalho, é indicado que este tipo de técnica gera um resultado bem mais simples de ser interpretado por humanos, porém existe uma dificuldade intrínseca de definir a saída correta de todos os dados de entrada a depender do problema.

Por se tratar de técnicas que entregam resultados concretos e mais simples de serem compreendidos, existem várias aplicações na literatura tanto no contexto de classificação, quanto no de regressão. Em **(Kotsiantis et al., 2007)** é realizado um estudo sobre diferentes técnicas de classificação, apresentando 5 principais classes de métodos de aprendizado para problemas dessa família: algoritmos baseados em lógica, algoritmos baseados em perceptrons, algoritmos estatísticos, algoritmos baseados em instâncias e algoritmos de máquinas de vetores de suporte.

Um exemplo largamente utilizado de algoritmos baseados em lógica são as árvores de decisão. Essa técnica consiste em dividir recursivamente o conjunto de dados baseado nas variáveis de entrada, uma de cada vez, sendo poderosa o suficiente para atuar em problemas tanto de classificação como de regressão **(Loh, 2011)**. Em cada nó da árvore é escolhida uma variável X dos dados de entrada para dividir o espaço de busca, e a melhor região de corte é

escolhida baseada em um critério denominado impureza que é calculado ao se verificar os filhos que possuem as classes de seus dados mais equivalentes entre si, ou seja, um nó filho que possui todos os dados com a mesma classe possui impureza zero enquanto um filho que possui metade de uma classe e metade de outra possui impureza um.

Os algoritmos baseados em perceptron são modelos baseados na abstração de sistemas biológicos proposta em **(Rosenblatt, 1958)**. As redes neurais surgiram a partir desse conceito, onde existem uma ou mais camadas de nós denominados de neurônios que possuem funções matemáticas internas que geram um resultado final discreto (classificação) ou contínuo (regressão). Essas funções matemáticas possuem pesos que precisam ser calculados a fim de tornar o modelo mais preciso, e o algoritmo que permite essa atualização dos pesos, que é conhecido como back-propagation **(Buscema, 1998)** atualiza os pesos através da diferença entre a saída desejada e a saída do modelo a partir do fluxo normal da rede neural com os pesos atuais.

Os modelos estatísticos, por sua vez, possuem uma base explícita de probabilidade e calculam a chance de dada entrada pertencer a uma determinada classe **(Kotsiantis et al., 2007)**. Um dos classificadores mais conhecidos desse grupo são os classificadores Naive Bayes, fundamentados na regra de Bayes da Probabilidade, onde o termo "Naive" vem do fato de ser necessário assumir que as características (features) de entrada são independentes entre si. Em **(Rish, 2001)** é levantado análises empíricas sobre o bom funcionamento desse algoritmo em problemas práticos, mostrando que, apesar de realizar essa suposição de independência, as decisões podem ser as corretas com certa frequência.

Segundo **(Aha et al., 1991)**, os algoritmos baseados em instâncias são modelos que utilizam instâncias específicas ao invés de abstrações pré-compiladas durante a etapa de previsão e, devido a manterem todos os dados de treinamento para a sua utilização, possuem um alto custo computacional durante a etapa de classificação. Um dos algoritmos mais conhecidos dessa classe é o de "K Nearest Neighbours", mais conhecido como KNN. Em (**Guo et al., 2003)** é definido como um simples algoritmo que depende do valor k que define quantos vizinhos ao padrão que pretende ser previsto serão levados em consideração e, para tal, é calculada uma distância, entre o padrão e todos os dados de treinamento. Esta distância pode ser a Euclidiana, Mahalanobis, Manhattan, dentre outras, sendo a qualidade da previsão muitas vezes vinculada à escolha do valor de k. Devido a essa dependência, existem estudos voltados a encontrar a melhor forma de determinar a quantidade otimizada de vizinhos, como em **(Zhang et al., 2017)** que propõe um algoritmo baseado em árvore que gera um valor de k não-fixo que se adapta melhor aos padrões de cada teste.

As SVMs ou máquinas de vetores de suporte, por sua vez, são algoritmos relativamente recentes e em **(Burges, 1998)** foi realizado um levantamento sobre o funcionamento dessa técnica. Com relação a modelos lineares, a ideia dessa técnica é gerar um híper plano de separação entre os dados, maximizando a distância entre os pontos de diferentes classes. No contexto de dados não-separáveis, são incluídos kernel e hiperparâmetros à técnica para tornar possível encontrar uma solução plausível.

A infinidade de métodos de aprendizagem supervisionada e problemas que podem atacar tornam necessário o vasto estudo sobre a melhor adaptabilidade a cada problema específico e o bom entendimento das características de cada modelo. Em **(Caruana & Mizil, 2006)** são comparados vários algoritmos, como KNN, SVM e redes neurais em 11 problemas de classificação binária, ou seja, possuem duas classes possíveis para cada entrada e, apesar de haver alguns métodos visivelmente superiores ou inferiores, existe uma grande variabilidade dependendo do problema e das métricas de avaliação escolhidas. Analogamente, **(Sivakumar et al., 2018)** propõe um algoritmo chamado *Mixed Mode Data Base Miner* (MMDBM) e compara o mesmo com outros 18 modelos de aprendizagem supervisionada sobre bases de dados de cancro de mama.

Um problema que os algoritmos supervisionados enfrentam (modelos de aprendizagem de forma geral) são os ruídos que podem existir nos dados. No estudo realizado por **(Nettleton et al., 2010)** são acrescentados diferentes níveis de ruídos tanto nos atributos sintéticos da base de dados como na classe de saída e é avaliado o impacto de 4 algoritmos de aprendizagem supervisionada, sendo o algoritmo Naive Bayes o mais robusto dentre as selecionadas.

Outro problema enfrentado pelos algoritmos supervisionados está relacionado à quantidade e qualidade de dados rotulados, ou seja, com a classe a qual pertence definido nos dados de treinamento. Em **(Zhou, 2018)**, o termo aprendizagem fracamente supervisionada é definida como sendo dividida em três tipos: dados que estão com as saídas rotuladas apenas parcialmente, dados com as saídas inexatas que corresponde a rótulos superficiais sobre os dados e dados com rótulos imprecisos que não são confiáveis por nem sempre serem verdadeiros. Por se tratar da complexidade em se obter dados precisos e em maior quantidade, esse conceito pode ser encontrado em inúmeros setores diferentes, como na área da saúde **(Tong et al., 2021; Bledowski, et al., 2021)**, computação gráfica **(Saito et al., 2021)**, biotecnologia **(Huang et al., 2021)**, onde enfrentam limitações técnicas e éticas para coleta de informação.

Algoritmos com base em processos biológicos estão em constante evolução com o objetivo de se aproximar da forma como os sistemas naturais aprendem, como, por exemplo, as próprias

redes neurais que são técnicas bastante difundidas, apesar de serem mais complexas de serem interpretadas. Em **(Wang et al., 2020)** é feito um levantamento de algoritmos conhecidos como redes neurais de spiking, que são redes neurais em que os conceitos de tempo no funcionamento do modelo são adicionados, onde um neurônio espera chegar a um certo limiar que faça com que dispare a informação para os outros neurônios, deixando claro que a utilização da mesma em larga escala depende da maior aplicabilidade e estudos no contexto de aprendizagem supervisionada.

Essas grandes infinidades de técnicas tornam claro a enorme quantidade de aplicações existentes em nosso cotidiano que podem ser construídas utilizando o contexto de aprendizagem supervisionada. Em (**Al Hasan et al., 2006)** os autores realizam um estudo sobre predição de conexão no escopo de redes sociais como uma tarefa supervisionada e, devido ao advento cada vez maior desses agrupamentos via internet, esse tipo de problema pode ser facilmente aplicado em outras aplicações de interação social e em outras aplicações como recomendação de link entre dois usuários. Em **(Carnevale et al., 2020)** ainda voltada a redes sociais, mas no ambiente de cuidado de saúde, é desenvolvido um estudo da utilização de classificadores para identificar publicações de pacientes críticos em redes sociais que contém informações de má qualidade ou mesmo erradas, sendo a aprendizagem supervisionada escolhida pela melhor aceitação na literatura científica dessa área.

Em **(Markovic et al., 2020)** é apresentada uma aplicação de modelos supervisionados na caracterização de viscosidade dos óleos de combustível vinculados ao processo de engenharia de recursos que constitui em tratar, transformar e retirar atributos a fim de melhorar a performance dos modelos, tanto em tempo quanto na qualidade de suas predições. Na área de previsão de riscos no contexto de combustíveis, o trabalho de **(Yin et al., 2021)** apresenta a aplicação de três algoritmos que buscam prever precocemente a diferença de pressão do gás durante as perfurações profundas de água com o objetivo de evitar graves problemas à segurança além de prejuízos econômicos desse processo industrial.

Outra área que se favorece bastante com os adventos da aprendizagem de máquina é a área médica. Em **(Kavuluru et al., 2015)** é analisada a aplicação de modelos na atribuição de códigos de diagnóstico a partir de registos médicos eletrônicos (EMR) de diferentes tamanhos, diferentes áreas médicas e número de códigos diferentes, onde prova-se que, quanto mais próximo do ambiente real de diagnósticos (grande e de diferentes especialidades), mais complexo fica para que os modelos sozinhos retornem resultados satisfatórios sem boas etapas de tratamento dos dados. Em **(Kim et al., 2019)** os autores apresentam a aplicação de aprendizado profundo em imagens médicas e discorrem sobre as

diferentes aplicações na área da radiologia, como classificação de imagens que consiste na verificação da presença ou não de doenças, deteção de objetos que consiste em encontrar uma região de interesse e caracterizar um objeto presente nela e segmentação de imagens que consiste em subdividir uma imagem em regiões menores que possuem as informações buscadas.

Os dados, por vezes podem trazer informações tendenciosas sobre o meio em que foram coletadas. Segundo **(Ghassami et al., 2018)**, sistemas de tomadas de decisão estão sendo cada vez mais utilizados em áreas como saúde, finanças e educação, e existe o risco de haver dados enviesados, que por vezes inserem discriminação em suas predições com base em dados sensíveis como orientação sexual, raça, religião, dentre outros. **(Hardt et al., 2016)** propõe um método para gerar predição levando em consideração um atributo de entrada considerado como sensível e utilizando ele para garantir que não haja discriminação, garantindo a não interferência direta do atributo protegido na saída predita pelo modelo e a solução foi testada sobre a previsão de capacidade de crédito sendo a raça um atributo sensível.

Ainda no contexto de equidade, **(Khodadadian et al., 2021)** analisa os efeitos do pré e pós processamento de dados com o objetivo de reduzir discriminação em saídas de algoritmos supervisionados sem deixar de garantir a acurácia do modelo, sendo em casos especiais um problema linear com uma solução eficiente em tempo. Também em **(Kamani et al., 2021)** é desenvolvido um trabalho semelhante que busca garantir uma boa relação entre equidade e acurácia do modelo com uma estratégia genérica para qualquer problema de otimização multiobjectivo baseada na fronteira Pareto com solução alcançada pelo método de Gradiente Descendente. Além disso, por se tratar de uma fronteira Pareto, é gerado um conjunto de soluções aceitáveis com diferentes níveis, relacionado a proporção de equidade versus acurácia, podendo ser escolhida a solução mais apropriada para cada aplicação.

De facto, essa é uma área rica em inúmeras possibilidades com relação a técnicas, problemas a serem abordados e consequências que a utilização dos seus modelos trazem para a sociedade. O uso consciente tanto dos dados, quanto dos modelos gerados a partir dos mesmos não é só necessário como também fundamental a fim de trazer real melhoria na vida das pessoas.

2.1.2. Aprendizado Não Supervisionado

Segundo **(Ghahramani, 2003)**, o aprendizado não-supervisionado corresponde a técnicas de aprendizado que recebem dados de entrada, porém não possuem acesso a nenhuma saída esperada e nem recompensas do ambiente por alguma descoberta, sendo responsável por encontrar padrões nos dados apenas com as informações intrínsecas a base de dados em si. Ainda segundo esse autor, esse tipo de aprendizagem geralmente está vinculada a modelos probabilísticos que podem ser usados em tarefas de classificação a partir da distribuição de probabilidades, deteção de outliers (dados da base que estão muito distantes do que poderia ser considerado como padrão), compressão de dados entre outras aplicações. Além disso, **(Hastie et al., 2009)** ressalta que o fato de não haver saídas desejadas bem definidas e não existir formas de medir o sucesso do modelo de forma direta torna mais complexo confirmar a validade de um modelo não-supervisionado.

Uma das técnicas mais conhecidas de aprendizagem não supervisionada são os algoritmos de agrupamento ou, clusterização. Segundo **(Serra & Tagliaferri, 2019)**, a clusterização é uma técnica exploratória cujo objetivo é unir dados semelhantes no mesmo grupo, levando em conta algum critério de distância e essas técnicas podem ser divididas em duas categorias: clusterização hierárquica e clusterização partitiva. **(Nielsen, 2016)** define a técnica hierárquica como sendo a construção de uma árvore binária com as folhas representando os dados de entrada e mesclando dois a dois os subconjuntos mais próximos um ao outro baseado no que é conhecido como distância de link. Diferente desse tipo, a clusterização partitiva ou de partições, busca gerar os grupos dos padrões de entrada simultaneamente sem gerar nenhum tipo de hierarquia **(Xiao e Yu 2012)**.

Um dos mais famosos algoritmos de clusterização partitiva é o algoritmo k-means. Esse modelo visa minimizar a distância entre cada ponto e o centro virtual do cluster (centróides) no qual faz parte. Em **(Hartigan e Wong 1979)** é apresentado o algoritmo descrito passo a passo, onde no início são definidos um número k de centros, onde k é uma entrada que representa a quantidade de grupos do modelo, e iterativamente realizar cálculos de distância entre todos os dados com todos os centros destes grupos, encontrando o grupo com menor distância. Após todos os dados serem separados, o novo centróide de cada grupo é calculado a partir do ponto médio gerado a partir dos padrões referentes àquele cluster.

Apesar de relativamente simples, esse algoritmo é incrivelmente de rápida convergência, porém ele garante apenas chegar em um mínimo local que depende diretamente dos centros escolhidos inicialmente e, devido a isto, existem estudos voltados a técnicas que inicializam

os centros de forma mais eficiente **(Yuan et al., 2004; Yedla et al., 2010)**. Outra preocupação é a seleção da quantidade de k que gerem grupos que realmente representem uma boa divisão e que seus indivíduos possuam semelhanças entre si. Em **(Pham et al., 2005)** descreve formas de seleção do valor de k, desde a entrada vinda de um usuário até medidas estatísticas, e propõe uma técnica baseada nas informações obtidas durante a operação do algoritmo de k-means.

Outro conjunto de técnicas de aprendizagem não-supervisionada são algoritmos de redução de dimensionalidade. Segundo **(Cunnigham, 2008)**, em algumas áreas, como análise de multimídia, existe um número enorme de recursos (*features),* o que pode tornar o processo lento e gerar uma performance baixa no processo de análise de dados, pois os algoritmos de aprendizado de máquina são suscetíveis a se degradarem com o aumento da quantidade de variáveis explicativas. Devido a isso, **(Cunnigham, 2008)** define que a redução de dimensionalidade tem como motivações: a identificação de um conjunto reduzido de *features* que gerem uma saída melhor para modelos posteriores; reduzir o tempo de treino e/ou classificação que tendem a aumentar com o número de atributos. Ruídos e *features* desnecessárias consideradas em modelos de predição reduzem a performance do mesmo e quanto maior a quantidade de atributos, mais difícil obter métricas de similaridade efetivas.

Um dos algoritmos mais conhecidos da literatura é o *Principal Component Analysis*, mais conhecido como PCA. Esse algoritmo permite transformar um conjunto de possíveis variáveis correlacionadas em um conjunto não-correlacionado com o objetivo de aumentar a variabilidade da base de dados **(Belarbi et al., 2017)**, permitindo a diminuição do espaço de busca.

Quando sozinhos, esses modelos entregam informações sobre os dados importantes, porém muitas vezes é complicado verificar sua utilidade, sendo muitas vezes utilizados como mecanismo anterior a aplicação de outros modelos posteriores. No trabalho de **(Jamal et al., 2018)** é comparada a aplicação do PCA e do *k-means* ambos na redução de dimensionalidade dos atributos de dados para predição de cancro de mama, dados esses que serviram de entrada em dois algoritmos de classificação e, mesmo que não seja a proposta inicial do *k-means*, este consegue obter boas performances. Em **(Cao et al., 2003)** é realizado um teste entre o PCA, o KPCA (PCA não-linear utilizando kernel) e o ICA na extração de features junto ao modelo SVM, demonstrando que essas técnicas podem ser usadas em conjunto a outras supervisionadas com um único objetivo.

Por outro lado, a não dependência de rótulos nessa classe de algoritmos possibilita que eles sejam utilizados em inúmeros contextos que só possuem os dados em si. Em **(Glielmo et al.,**

2021) é apresentada a utilização de métodos não-supervisionados para simulação de moléculas permitindo construir dados mais compactos, porém efetivos na determinação da trajetória molecular. Além dos modelos de redução de dimensionalidade e de agrupamento, são apresentados também modelos cinéticos que são algoritmos aplicados a diminuição das variáveis que levam em conta a correlação gerada pela inclusão de tempo nos modelos. Em **(Rives et al., 2021)** é desenvolvido um estudo com o intuito de aplicar a etapa não-supervisionada de transformers, modelo de aprendizado profundo que pondera cada etapa de entrada, sobre 86 bilhões de aminoácidos de 250 milhões de sequências a fim de aprender sobre propriedades biológicas intrínsecas, o que obteve recursos não existentes no estado da arte da área e que podem ser combinados com modelos preditivos para melhorar os resultados da metodologia.

Essa característica no contexto de imagens gerou um programa desenvolvido em **(Wu et al., 2021)** conhecido como aprendizagem progressiva não-supervisionada a partir de vídeos não rotulados para rastrear objetos seguindo 3 etapas: aprendizagem de discriminação de fundo, usando técnica de mineração negativa baseada em âncora de imagens aumentadas, mineração temporal, aplicando o modelo de discriminação de fundo para minerar os componentes temporais e, por fim, aprendizagem de correspondência temporal a partir de uma função de perda robusta ao ruído. Em **(Ho et al., 2021)** também se utiliza aprendizagem não-supervisionada na recuperação da estrutura 3D de imagens não-simétricas aplicando perdas de consistência de forma e textura para fazer o modelo convergir apresentando ótimos resultados qualitativos e quantitativos.

Tendo essas aplicações em vista, pode-se perceber a importância de técnicas não-supervisionadas para entender os dados existentes e podem também trazer melhorias de performance não só através da redução de dimensionalidade, como com a criação de novas *features*. Em **(Piernik e Morzy, 2021)**, foi realizado um extenso estudo para avaliar como o uso de agrupamento na criação de novas *features* possibilitaria uma melhoria na performance da predição do modelo. Os experimentos avaliaram dois algoritmos de agrupamento, k-means e agrupamento por afinidade **(Frey e Dueck, 2007)**, o uso de cinco técnicas de codificação dos clusters em novos atributos, entre outros cenários. Os resultados obtidos indicaram que adicionar atributos criados por meio de agrupamentos pode melhorar a qualidade da modelagem, porém o modelo de aprendizagem escolhido e o método de codificação das novas *features* influenciam no resultado.

Sem o problema causado por informações imprecisas quanto aos rótulos, o desafio que a aprendizagem não-supervisionada enfrenta reside na imprecisão dos dados em si, tanto devido

a ruídos como erros que podem gerar os famosos outliers, que são pontos de dados muito discrepantes da realidade e distribuição dos dados **(Hawkins, 1980)**. Com isso, existem formas de lidar com outliers baseadas em modelos não-supervisionados como foi abordado em **(Campos et al., 2016)** que apresenta um estudo elaborado não só sobre a prática necessária de identificar os outliers, como também de se verificar os problemas existentes tanto nas técnicas como nas metodologias de avaliação. Para tal, (**Campos et al., 2016)** utiliza os algoritmos de KNN no contexto de deteção de ruídos devido a popularidade desses métodos na literatura. Os comportamentos dos diferentes métodos podem ser comparados tomando k como base, mesmo que o k possua interpretação diferente a depender do modelo e são testadas 12 técnicas, em 23 conjuntos de dados utilizando várias métricas de análise de performance e chega-se a conclusão de que não faz sentido expressar que uma técnica é melhor do que a outra apenas a partir uma única métrica de avaliação. Além disso, foi possível demonstrar a ampla gama de possibilidades na simples análise de conjunto de dados e métricas de avaliação no âmbito de remoção de ruídos.

2.2. NATURAL LANGUAGE PROCESSING

Segundo **(Chowdhury, 2005),** o Processamento de Linguagem Natural, comumente conhecido como NLP, é uma área de pesquisa e exploração de mecanismos que possibilitam a manipulação de texto falado e escrito pelos computadores. Uma definição mais formal que denota o termo NLP em **(Liddy, 2001)** expressa que NLP é um grupo de técnicas computacionais para analisar e representar naturalmente um ou mais níveis de análise linguística a fim de alcançar uma aparência humana no processamento da língua em várias tarefas e aplicações. De facto, a língua é um mecanismo bastante variado, dependente da geografia e vasto para ser facilmente compreendido pelas máquinas, gerando um interesse não só na reprodução no interior dos computadores para compreensão do que se é dito, mas também na reprodução da língua em aplicações, como os famosos chatbots.

Sendo uma área em constante ascensão, é apresentado em **(Hirschberg e Manning 2015)** alguns dos avanços aplicados a produtos de consumo direto como a Siri da Apple, justificados por quatro fatores principais: aumento do poder computacional, a disponibilidade de grande quantidade de dados de texto (livros, artigos, redes sociais), o desenvolvimento de algoritmos de aprendizagem de máquina cada vez melhores e uma compreensão maior sobre a estrutura da linguagem humana e sua aplicação em contextos sociais. No campo da tradução via máquina o objetivo dos algoritmos é não apenas entender a ordem e significado das palavras, mas como o contexto influencia uma frase e conhecimento de mundo, além de a compreensão

de discurso ter sido bastante aprimorada **(Hirschberg e Manning 2015)**. No contexto dos mecanismos de busca, o aprimoramento do conhecimento da linguagem tornou possível a criação de mecanismos que compreendem consultas mais complexas, mesmo que os resultados sejam de outro idioma.

A partir do trabalho desenvolvido em **(Vaswani et al., 2017)**, muita coisa mudou em NLP. Este artigo deu a fundamentação aos Transformers, que se trata de uma arquitetura de modelo simplificada que evita o processo de recorrência, confiando no que é chamado de mecanismo de atenção, que por sua vez calcula dependências entre entradas e saídas de cada termo separadamente. Isto torna essa arquitetura bastante suscetível a paralelização e, com isso, gera ganhos de performance e possibilita utilizar bases de dados maiores. Rapidamente essa arquitetura ganhou o status de estado de arte tanto na área de visão computacional, mas principalmente na área Processamento de Linguagem Natural. Em **(Wolf et al., 2020)** descreve o funcionamento da biblioteca *Transformers*, mantida por engenheiros da Hugging Face que possui três blocos em sua formação: o primeiro é o *Tokenizer* responsável por transformar texto puro em codificações de índices denominado tokens, o segundo é um *Transformer* que capta esses tokens e transforma em um vetor contínuo para cada palavra conhecido como embedding contextual e, por fim, o terceiro conhecido como *Head* que utilizam os embedding para realizar predições específicas para a tarefa.

Novos modelos de linguagem foram surgindo utilizando a arquitetura de *Transformers*. **(Devlin et al., 2018)** introduz *Bidirectional Encoder Representations from Transformers*, conhecido como BERT, modelo desenvolvido para pré-treinar representações bidirecionais a partir de texto não-rotulado, resultando em um modelo que pode ser incrementado com uma camada de saída para criar modelos voltados a resolução de uma tarefa, ou seja, não precisa modificar o modelo pré-treinado para aplicar em diferentes tarefas da área de NLP. De facto, o BERT representa o primeiro modelo baseado em ajuste fino, ou fine-tuning, que atinge desempenho em grande quantidade de tarefas mesmo sem ser construído especificamente para uma tarefa, sendo pré-treinado com documentos do Wikipedia Inglês e BooksCorpus, possuindo cerca de 3300M palavras no total. Foram realizados 11 procedimentos de ajustes finos referentes a 11 problemas de NLP envolvendo respostas a perguntas, predizer continuação de texto, dentre outras atividades e foram testados diferentes tamanhos do modelo que obtiveram performances consideráveis em todas tarefas escolhidas para teste.

O pioneirismo do BERT e seu sucesso abriu margens para a criação de variantes dele com o objetivo de torná-lo mais performativo e/ou eficiente. Em **(Liu et al., 2019)** é apresentado o modelo RoBERTa que é gerado a partir de uma metodologia de treinamento sobre modelos

do BERT com as seguintes modificações: treinar os modelos durante mais tempo e com pedaços de dados de treinamentos maiores, remover a predição de próxima sentença, treinar em sequências maiores e mudar de forma dinâmica o padrão de máscara aplicado nos dados de treinamento. Além disso, foi utilizado uma nova base de dados de treinamento, confirmando com isso que mais dados colaboram com o aumento da performance do modelo. Os resultados do modelo foram importantes, tendo superado os algoritmos estado da arte em 4 das 9 tarefas do GLUE (*benchmark* utilizado na área de NLP) e obteve melhores resultados que o BERT em todas as tarefas, provando que o cuidado com o design do modelo e a quantidade de dados utilizado pode trazer grandes melhorias no BERT.

Outra variação conhecida do BERT é o algoritmo proposto por **(Sanh et al., 2019)** conhecido como DistilBERT, uma versão menor, mais rápida e de menor custo computacional. Mesmo treinando um modelo de linguagem menor, foi possível manter 97% da compreensão de linguagem do BERT, reduzindo seu tamanho em 40% e tornando 60% mais rápido. A motivação por trás desse empenho em diminuir o modelo é de melhorar o tempo de inferência do mesmo, diminuir o custo de armazenamento, mais rapidez no treinamento, sem prejudicar a qualidade de suas saídas. O nome DistilBERT vem do termo *Knowledge distillation* que é uma técnica de compressão que permite a um modelo compacto treinar para reproduzir o comportamento de um modelo maior.

Outro modelo recente desenvolvido a partir dos Transformers é o Generative Pre-trained Transformer, mais conhecido como GPT, sendo a terceira geração desse algoritmo sua última versão. O intuito desse modelo, segundo **(Floridi e Chiriatti, 2020)**, é gerar sequências de palavras ou outros dados começando a partir de uma entrada e, devido a esta complexidade, é necessário realizar treinamento não-supervisionado em uma quantidade enorme de dados. De facto, em **(Brown et al., 2020)** é descrito que o GPT-3 foi treinado utilizando 175 bilhões de parâmetros, cerca de dez vezes maior que qualquer outro modelo, e ele é aplicado diretamente a tarefas sem ajuste fino, obtendo excelentes performances com poucas entradas a partir de interações diretas com o modelo. Com isso, esse modelo tem aplicabilidade em diversas tarefas, como tradução, sumarização, correção gramatical, resposta a perguntas, chatbots entre outras **(Floridi e Chiriatti, 2020)**.

De facto, existem inúmeras tarefas existentes na literatura relacionadas ao estudo de Processamento de Linguagem Natural. As máquinas de tradução são uma das aplicações mais antigas da Ciência da Computação e que possui bastante interesse comercial, onde o objetivo é realizar a tradução automática de uma língua para outra. Segundo **(Specia e Wilks, 2021)**, essa tarefa possui quatro abordagens principais: máquina de tradução baseada em um

conjunto de regras estabelecidas que atuam em diferentes níveis linguísticos, máquina de tradução baseada em exemplos que utiliza de traduções de textos existentes na base de dados para construir traduções de novos textos, máquina de tradução estatística que extrai regras de tradução criando um modelo a partir dessas regras e as máquinas de tradução híbridas.

Ainda no contexto de tradução, **(Han et al., 2021)** apresenta uma metodologia de aplicação de modelos generativos de linguagem na tradução sob aprendizagem não-supervisionada sobre o problema de tradução entre Inglês e Português. Esse método consiste em três etapas: few-shot amplification, distillation e backtranslation. A few-shot amplification amplifica traduções de frases não rotuladas utilizando alguns dados para amostragem de uma grande base de dados sintética, sofrendo distallation que é o processo onde há o descarte das demonstrações, sendo realizado um ajuste fino e, por fim, durante o backtranslation são geradas repetidamente traduções para um conjunto de entrada ajustando um único modelo garantindo a consistência cíclica **(Han et al., 2021)**.

O problema de sumarização em NLP está relacionado à habilidade da máquina de entender o texto e extrair as informações principais presentes nele, gerando uma saída plausível sobre o texto. Em **(Kumar e Rani, 2021)**, é proposto um algoritmo de sumarização baseado na frequência das palavras do texto e métodos da biblioteca NLTK, para extrair os pontos principais do texto de entrada, onde o usuário entra com o texto, é realizada a remoção temporária de Stop Words, a aplicação de stemmer e a tokenização de palavras como pré-processamento do texto. Cada sentença irá receber uma pontuação e é encontrado um limiar baseado no número de textos do algoritmo, após esse processo é calculada a média da pontuação de cada sentença, onde os pontos principais são extraídos baseados no limiar e, por fim, é gerada uma saída.

(Awasthi et al., 2021) apresenta uma pesquisa voltada a sumarização baseada em texto e categoriza essa tarefa como sendo extrativa ou abstrata, sendo métodos extrativos aqueles que retiram um subconjunto de sentenças do texto original e os métodos abstratos são os que criam a sumarização utilizando palavras que não estavam no texto de entrada. Além dessas classes, cada categoria possui exemplos de utilização de técnicas da aprendizagem supervisionada e não-supervisionada, abrindo margem para métodos bastante distintos. Em **(Bondielli e Marcelloni, 2021)**, por sua vez, é utilizado o conceito de sumarização juntamente com a arquitetura de Transformer e algoritmos de clusterização para realizar a criação de perfis de currículos aplicada em uma base de domínio público. Nesse contexto, a sumarização é utilizada como forma de reduzir os currículos de forma a diminuir redundância e melhorar a representação dos dados.

Uma das vantagens obtidas com o avanço do Processamento de Linguagem Natural é a possibilidade cada vez maior de sistemas inteligentes interagirem com o usuário diretamente tanto na resolução de problemas quanto respondendo a dúvidas, sem a interação de outro ser humano. Em **(Yin et al., 2019)** é desenvolvido uma ferramenta para resposta a perguntas sobre visualização e análises espaço temporais, com o intuito de permitir que usuários possam ministrar essas análises com menor carga cognitiva facilitando a interface entre usuário e os insights. A proposta do trabalho foi desenvolver uma arquitetura com micro serviços onde o usuário pudesse digitar ou falar suas perguntas, realizando testes sobre dados climáticos e permitindo um retorno automático de análises que seriam complexas para um usuário executar com pouco conhecimento de domínio.

Como forma de promover a aproximação entre cuidados de saúde e moradores do campo, foi proposto em **(Vinod et al., 2021)** um pipeline de NLP de um sistema de resposta a perguntas que permita o acesso à informação na área rural resolvendo problemas durante o primeiro contato entre médico e paciente. Um outro fator motivacional do trabalho foi a dificuldade do acesso à informação da área rural em tempos de COVID-19. Foram utilizados modelos de linguagem derivados do BERT e o GPT-2 em seus testes obtendo resultados promissores em 3 bases de dados existentes de Question Answering (QA).

Com o advento dos chatbots e a qualidade dos mecanismos que os geram, a comunicação com os clientes de uma empresa tem sido substituída cada vez mais por agentes conversacionais. Apesar de trazer velocidade nas respostas, nem sempre as respostas agradam o usuário. Em **(Adam et al., 2021)**, é realizado um estudo da utilização de chatbots e os efeitos que isso traz para a experiência do usuário e a aceitação do mesmo às ferramentas de inteligência artificial conversacional. Os resultados do estudo comprovaram que mesmo que os usuários soubessem que estavam conversando com chatbots, eles ainda respeitavam as mesmas regras sociais. Outra contribuição foi a descoberta de uma relação entre as técnicas de comunicação utilizadas e um certo nível de obediência dos usuários com relação às interações com o bot. Outro estudo no contexto de chatbots foi proposto em **(Melián-González et al., 2021)** que foi focado na análise da intenção dos usuários ao entrar em contato via agentes inteligentes com empresas do ramo de viagem e turismo. O estudo indica que existem inúmeros fatores que levam os usuários a falarem diretamente com os bots, como disposição em usar tecnologias self-service, influências sociais e mesmo a qualidade do bot em simular um humano.

Em atividades muito específicas, mesmo os modelos robustos como o BERT precisam passar pela etapa de pré-treino utilizando documentos específicos do escopo para melhorar a performance dos modelos linguísticos. Na área da biomedicina, **(Gu et al., 2021)** propõe

realizar treinamento do zero, a fim de obter ganhos na compreensão de documentos específicos de domínio a partir de dados não rotulados da área. De facto, a ideia por trás desse trabalho é verificar a hipótese de que documentos de domínio geral são tão diferentes dos textos envolvendo biomedicina que acabam atrapalhando a performance do modelo na tarefa escolhida. De facto, após realizar os testes, o modelo gerado a partir do treinamento com documentos do domínio específico superou os modelos do BERT e RoBERTa em todas as tarefas com uma considerável margem de diferença.

Nas diversas aplicações de NLP, um dos problemas que colabora com a boa solução de um modelo é conhecido como reconhecimento de entidades nomeadas, ou NER. Esse problema é responsável por localizar nomes importantes ou substantivos próprios no texto, como nomes de cidades/países, nomes de empresas, dentre outros **(Mohit, 2014)**. De facto, em tarefas de máquinas de tradução, por exemplo, a palavra Apple poderia ser facilmente traduzida como maçã, porém, tanto pela palavra iniciar com maiúscula, como pelo contexto, poderíamos inferir quando esta palavra é associada à empresa. Em **(Weber et al., 2021)** é apresentada uma ferramenta de NER de fácil uso voltada para aplicações na área biomédica, sendo algo necessário. No contexto da biomedicina, nomes de doenças, químicos ou de genes acabam sendo importantes, e extrair informações semanticamente relacionadas com elas se torna uma tarefa relevante, porém a especificidade da área torna outras ferramentas de NER pouco performativas em captar entidades em certos termos.

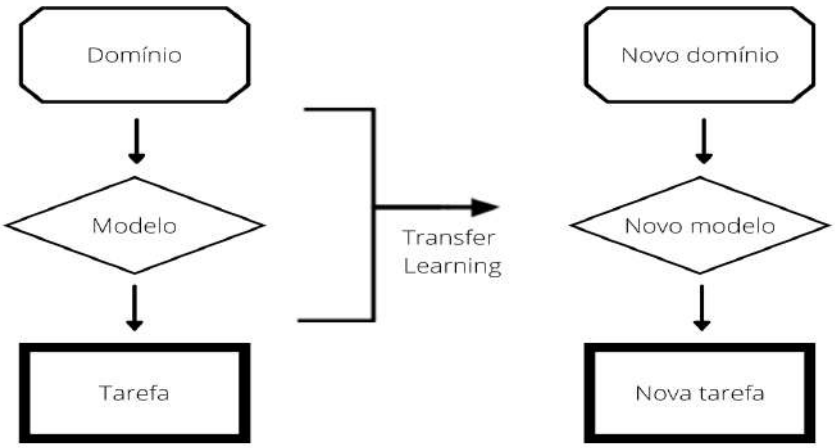
Ratificando essa inabilidade que algumas ferramentas pré-treinadas têm com relação ao NER, em **(Yang et al., 2021)** é desenvolvido um estudo para encontrar entidades em relatórios de vulnerabilidade de segurança realizando fine-tuning dos algoritmos BERT, RoBERTa e Electra, sendo possível encontrar resultados tão bons quanto os pré-existentes na literatura com apenas 11,2% da base de dados. Em **(Alonso, 2021)** foi aplicado um ajuste fino sobre BioBERT (uma variação de BERT voltado para a área da biomedicina) que foi utilizado em dois casos práticos envolvendo textos da web e artigos específicos relacionados ao SARS-CoV-2, obtendo a conclusão de que essa técnica com pouco esforço se adaptou bem na tarefa de NER sobre esses artigos com um tema tão recente.

2.3. TRANSFERÊNCIA DE APRENDIZADO (*TRANSFER LEARNING*)

Uma definição formal de *Transfer Learning* é encontrada em **(Yang et al., 2020)** que apresenta o processo de transferência de aprendizado como uma técnica de melhorar uma tarefa preditiva nova T_t a partir de um domínio alvo D_t, utilizando de conhecimento adquirido a partir de um modelo gerado para solucionar uma tarefa T_s diferente em um domínio D_s

também distinto. Na Figura 1 a seguir, pode-se observar o fluxo básico do processo de Transfer Learning, onde o lado esquerdo representa o processo natural de desenvolvimento de *machine learning* e no lado direito o resultado da transferência de aprendizado.

Figura 1 - Processo de Transfer Learning



Fonte: Próprio Autor

Essa metodologia tem alguns objetivos como exposto em **(Yang et al., 2020)**:

- Muitas aplicações possuem poucos dados disponíveis e técnicas tradicionais de aprendizagem de máquina não são suficientes para generalizar em novos cenários.
- Modelos de Machine Learning precisam ser robustos e, em situações em que os dados de treinamento e de teste não possuem a mesma distribuição e/ou variam com o tempo, sendo necessários outros métodos de adaptação dos modelos.
- Em problemas do mundo existe pouco acesso a dados pessoais por motivos de privacidade e segurança, tornando complicado para modelos de aprendizagem de máquina proporcionar resultados mais personalizados a uma situação específica.
- Em situações para manter a privacidade dos dados, espera-se extrair apenas os pontos mais importantes dos dados e utilizar apenas isso no modelo, de forma a não trafegar e utilizar dados muito sensíveis e específicos do usuário.

Esse processo de *Transfer Learning* tem sido um ponto bastante abordado devido as necessidades descritas acima, gerando muitos estudos envolvendo NLP. Em **(Ruder et al., 2019**) é realizado um estudo geral sobre a utilização de transferência de aprendizado no contexto de processamento natural de linguagem, principalmente se baseando em *sequential*

transfer learning, que consiste numa fase de treinamento prévio onde as representações gerais de uma tarefa fonte são apreendidas e uma segunda fase de adaptação responsável por aplicar o conhecimento anterior na nova tarefa do novo domínio. Voltado a um problema bem atual e recorrente, em **(Mozafari et al., 2019)** é aplicado o conceito de *Transfer Learning* baseado no BERT para detetar discurso de ódio em redes sociais realizando uma combinação do modelo BERT pré-treinado sem supervisão e algumas técnicas supervisionadas de fine-tuning. Por ser um domínio bastante específico, a ideia principal foi reaproveitar conhecimento generalista de um modelo pré-treinado no escopo de deteção do discurso de ódio, obtendo resultados mais performativos que os existentes na literatura e, além disso, foi possível verificar nos resultados uma capacidade de identificar vieses no processo de coleta de dados, favorecendo assim a equidade do modelo, conhecido também como *fairness.*

Com relação a problemas onde é necessário observar o contexto e a compreensão de um domínio pode auxiliar em uma tarefa de outro domínio, como o problema de reconhecimento de entidades nomeadas, é possível aplicar o conceito de Transfer Learning. Em **(Francis et al., 2019)** é realizado um estudo da aplicação dessa metodologia em documentos financeiros e de biomédicos, demonstrando com os experimentos que um modelo NER treinado em dados rotulados podem ser usados como modelos base para serem utilizados em um outro domínio com alguns ajustes, usando poucos dados do domínio de destino e, para tal, utilizou-se mecanismos de atenção e modelo BERT para criar variações de modelos a serem analisados. Em **(Sun e Yang, 2019)**, também foi desenvolvido um estudo sobre a aplicação de Transfer Learning sobre o NER em domínio da Biomedicina, utilizando dois modelos multilíngues e foi possível verificar que conhecimento de domínio mais geral são efetivas na aplicação de Transfer Learning.

2.4. PRÉ PROCESSAMENTO DE DADOS PARA NATURAL LANGUAGE PROCESSING

O pré-processamento de dados é uma das etapas mais importantes em um projeto de análise de dados. Este é um conjunto de tarefas que envolve a preparação, limpeza, organização e estruturação dos dados, para que estes possam ir de dados brutos para dados prontos para utilização.

Alguns problemas comumente encontrados e que são solucionados na etapa de pré processamento de dados são: dados em branco, dados com ruídos (geralmente dados que podem ser utilizados, mas precisam de alguma alteração) e valores discrepantes (outliers).

Nesse tópico serão abordadas as principais técnicas de pré-processamento de dados no contexto de NLP que são a tokenização, a capitalização, a remoção de stop words, o

stemming e o lemmatization, passando por suas definições, aplicações, desafios e inovações que existem em cada uma delas.

2.4.1. Tokenização (Tokenization)

O processo de tokenização é uma das etapas mais importantes e primordiais do processamento de linguagem natural. Em **(Webster e Kit, 1992)** esse processo é caracterizado como inicial no pré-processamento e é definido como a etapa responsável por dividir a entrada em unidades básicas, conhecidas como tokens, que não precisam ser decompostas em etapas posteriores para NLP.

De facto, **(Grefenstette, 1999)** ratifica a importância dessa etapa e são levantados alguns casos que mostram a complexidade da tarefa, pois devem haver cuidados extras ao realizar a tokenização de palavras de nomes próprios, abreviaturas, dentre outros exemplos. Outro ponto de atenção seria a ambiguidade gerada por esse processo, uma vez que as palavras são transformadas em estruturas mais básicas perdendo informação de contexto e conhecimento do mundo em si.

Além disso, **(Mullen et al., 2018)** relata que esta tarefa é computacionalmente custosa, dependente da linguagem que será processada e os problemas dessa etapa influenciam os resultados do modelo com um todo. Isso torna bastante relevante o estudo sobre a tokenização dos textos de entrada em várias línguas diferentes, como coreano **(Park et al., 2020)**, estoniano **(Orasmaa et al., 2016)**, inclusive ferramentas para multilingual **(Agerri et al., 2014.; Garcia e Gamallo, 2015.)**, e também algoritmos mais performativos de tokenização **(Song et al., 2020; Rust et al., 2020)**.

O BERT **(Devlin et al., 2018)**, algoritmo que ganhou destaque desenvolvido por uma equipe do Google, utiliza um mecanismo de tokenização baseado em dados visando maximizar a probabilidade do modelo dos dados de treinamento. Cada palavra pode ser quebrada em subpalavras dado um conjunto de textos, conhecido como corpus, e um número desejado de tokens. No trabalho em questão, foi verificado que um vocabulário entre 8000 e 32000 tokens atingiam uma performance adequada e mantiveram uma rápida velocidade de decodificação entre todos os pares de palavras.

2.4.2. Capitalização (Capitalization)

O termo capitalização, também conhecido como "*truecasing*", refere-se à utilização de letras maiúsculas (*capital letters*) em determinado texto e **(Păiş e Tufis, 2021)** indica que o foco está na identificação da forma das palavras, sendo subdivida em quatro classes: todas as letras

minúsculas, todas as letras maiúsculas, apenas a primeira letra maiúscula e uma mistura de letras maiúsculas e minúsculas.

Segundo **(Carey, 1980)**, marcadores gramaticais, como a capitalização, podem ser utilizados para transmitir sentimentos de stress e modificação no tom do texto. De facto, em **(Km et al., 2015)** são apresentados resultados que demonstram que a análise da quantidade de letras maiúsculas em uma sentença gerou resultados mais assertivos quanto à deteção de emoções.

(Chan e Fyshe, 2018) dá o exemplo dos acrônimos para demonstrar que nem sempre a capitalização tem um significado na identificação emocional, sendo necessário utilizar técnicas que identifiquem quando uma letra maiúscula tem uma importância semântica para o texto. Além disso, com o advento das redes sociais, o coloquialismo, o neologismo e mesmo erros gramaticais aumentam ainda mais a necessidade de analisar com cuidado se a grafia das palavras é informativa ou não **(Nebhi et al., 2015)**.

Com isso, essa característica pode ser usada em muitas aplicações do Processamento de Linguagem Natural. No âmbito de análise da importância da capitalização, **(Pak e Teh, 2018)** apresenta o significado da utilização das letras maiúsculas na revisão de produtos, onde é constatado que o seu uso implica em uma maior ênfase do sentimento associado à opinião, seja negativa ou positiva. **(Zhang et al., 2021)** apresenta um modelo baseado em redes neurais para identificar o "truecasing" em uma palavra que ajuda a melhorar a performance de outras tarefas de NLP, como reconhecimento de entidade nomeada, ou NER. Outra aplicação foi apresentada em **(Mayhew et al., 2020)**, onde foi desenvolvido um modelo para identificar se uma palavra deveria ser maiúscula ou minúscula melhorando a performance de algumas bases de dados também para o NER.

2.4.3. Palavras Irrelevantes (Stop Words)

Segundo **(Wilbur e Sirotkin, 1992)**, Stop Words são termos que possuem função gramatical, porém não fornecem informação relevante sobre o conteúdo de documentos e sua remoção pode trazer benefícios na análise e recuperação do mesmo. Ainda nesse trabalho, é proposto uma abordagem estatística onde é calculada similaridade entre pares de documentos como sendo o suficiente para determinar se uma palavra é uma *stop word* ou não.

Assim como na capitalização, esse tipo de termo é dependente da linguagem dos documentos, tornando necessário estudo em diferentes idiomas. Em **(El-Khair, 2006)** são apresentados resultados de extração de informações após a remoção de *stop words* de documentos de idioma árabe utilizando três listas de palavras: uma genérica da língua, uma baseada no corpus de entrada e uma combinação de listas. Por sua vez, **(Hao e Hao, 2008)** faz uma

proposta de algoritmo para identificar automaticamente *stop words* na língua chinesa onde estas devem atender duas condições: terem uma alta frequência nos documentos e possuir baixa correlação com todas as categorias de classificação.

Devido ao amplo estudo nesse contexto, existe o esforço na criação de listas de palavras que podem ser excluídas sem perder informação relevante dos documentos. Em **(Bird et al., 2009)** é mostrado o processamento de linguagem natural utilizando a biblioteca Natural Language Toolkit, mais conhecida como NLTK, que possui implementadas listas de palavras baseado em corpus como recurso léxico, incluindo as *stopwords* de diferentes linguagens. Porém, em alguns contextos específicos essa lista deve ser identificada diretamente por documentos pela alta frequência da presença de certos termos específicos da área. **(Sarica e Luo, 2021)** apresenta um estudo de stop words em documentos técnicos utilizando patentes com milhões de sentenças no total e adicionando à lista encontrada ao pacote padrão do NLTK em conjunto com a lista USPTO (lista prévia com termos de patentes), sendo possível obter melhores resultados ao remover os termos presentes nas três listas.

De facto, em **(HaCohen-Kerner et al., 2021)** foi feito um estudo de diversas técnicas de pré-processamento em tarefas de classificação textual e a simples remoção das *stopwords* apresentou resultados satisfatórios, mesmo que tenha obtido resultados irrelevantes em algumas das bases de dados. Por conta da sua importância, a remoção de termos frequentes é alvo de estudo em ótica multilingue, porém isso vem com um alto custo de treinamento, pois tende-se a usar uma quantidade imensa de textos de entrada. Em **(Ferilli, 2021)** apresenta uma proposta de identificação automática desses temos utilizando poucos documentos, até mesmo únicos, pois, segundo o autor, grandes quantidades de documentos de entrada não são simples de coletar em todas as línguas e seria uma técnica que não precisaria de conhecimento linguístico prévio. Como resultado, houve uma grande taxa de acerto do que as técnicas presentes na literatura até então.

2.4.4. Stemming

Em idiomas morfologicamente ricos, a quantidade de variações morfológicas ao redor de uma mesma palavra base tende a acontecer com frequência e, devido a essa característica, é necessária a aplicação de técnicas de stemming com o objetivo de simplificar essas variações **(Singh e Grupta, 2016)**. Por exemplo, as palavras “gostar”, “gosto” e “gostoso” possuem a mesma palavra base. Ainda neste trabalho, é apresentado um vasto estudo sobre as formas de aplicar o processo de stemming em um documento, os métodos em diferentes idiomas e suas várias aplicações em domínios diferentes, como classificação e agrupamento de documentos,

sistema de sumarização automática, sistemas de tradução, sistemas de resposta a perguntas, dentre outros.

(Paice, 1994) cita problemas no processo de stemming como sendo de duas classes: *overstemming* e *understemming*. Ambos os problemas estão relacionados aos conceitos das palavras, sendo que o *overstemming* ocorre quando duas palavras com raízes distintas e conceitos distintos são identificadas com a mesma base, já o *understemming* ocorre quando duas palavras com a mesma raiz e conceitos equivalentes são classificadas com bases diferentes.

Por ser uma etapa complexa e suscetível a esse tipo de falha, existem vários tipos de algoritmos responsáveis pela extração do *stemming* das palavras e, segundo **(Jivani, 2011)**, podem ser classificadas em três grupos: algoritmos de truncamento, algoritmos estatísticos e algoritmos mistos. Os algoritmos de truncamento são responsáveis pela extração de sufixos e prefixos de uma palavra a partir do corte das mesmas, os algoritmos estatísticos funcionam após a aplicação de algum procedimento estatístico para identificação dos afixos, já os algoritmos mistos possuem particularidades na base de dados das palavras utilizados no seu processo, como um *stemmer* baseado em corpus.

Como dito anteriormente, línguas ricas morfologicamente tendem a apresentar maiores variações e, consequentemente, precisam de maiores cuidados no processo de stemming. **(Nzeyimana, 2020)** apresenta um estudo sobre retirada de ambiguidade contextual de formas verbais na língua kinyarwanda, falado principalmente em Ruanda, e que não possui tantas ferramentas próprias para se trabalhar com esse idioma. Por sua vez, em **(Rianto et al., 2021)** é desenvolvido uma melhora nessa transformação dos termos sob o contexto de conversas não-formais no idioma da Indonésia e, apesar dos bons resultados nos documentos que tinham acesso, o algoritmo desenvolvido possui algumas limitações quanto a automaticidade do processo e incompletude das palavras raiz, reforçando a complexidade e importância do tema no pré-processamento em NLP.

2.4.5. Lemmatization

Apesar do processo de lemmatization ser igualmente morfológico, trata-se de um processo de simplificação diferente do Stemming. **(Plisson et al., 2004)** descreve o processo como uma normalização onde, ao invés de procurar um radical para as palavras, é realizado um processo de normalização, gerando uma palavra simplificada de mesmo contexto à sequência de palavras que precisam ser normalizadas.

No exemplo anterior das palavras “gostar”, “gosto” e “gostoso”, o processo de stemming geraria o radical “gost” o que, em alguns contextos, pode gerar ambiguidade como dito anteriormente. Já no caso da lemmatização a palavra normalizada poderia ser “gostar”, dessa forma existe uma palavra compreensível e com contexto próprio como retorno, ao invés do radical unicamente. Por ser um processo mais complexo, **(Khyani et al., 2020)** define que os resultados da utilização de lemmatization são mais precisos, mas é necessário um maior conhecimento na criação de vocabulários adequados.

Devido a suas características e aplicações semelhantes por se tratar de um pré processamento morfológico, ambas técnicas são vastamente estudadas em conjunto a fim de realizar sua aplicabilidade e mesmo comparação entre elas. **(Balakrishnan e Lloyd-Yemoh, 2014)** apresenta um estudo que compara lemmatization e stemming na recuperação de documentos e um algoritmo que não utiliza processamento de linguagem, que mostra a melhoria da performance em relação ao modelo de teste sem o processamento e da lemmatization sobre o stemming. Tal comportamento pode ter sido causado, segundo o autor, por se tratar de uma técnica mais avançada que pode levar em conta até sinônimos das palavras, podendo trazer resultados mais semelhantes.

De forma análoga, **(Korenius et al., 2004)** aplicou ambas técnicas no agrupamento de documentos finlandeses para comprovar a hipótese de que a lemmatização teria uma performance melhor devido às características do idioma. Foram feitos testes em quatro métodos de agrupamento de documentos e, apesar de em alguns testes uma das métricas apontar resultados melhores utilizando stemming, a lemmatization obteve resultados mais balanceados o que gerou uma busca mais eficaz na maior parte das análises e é apontado que uma vantagem desta esteja na possibilidade de lidar com palavras compostas do idioma finlandês.

Devido a sua complexidade, existem vários estudos com o objetivo de tornar o processo menos dependente de conhecimento linguístico prévio. **(Akhmetov et al, 2020)** apresenta um modelo de classificação com árvores de decisão com o intuito de melhorar a performance na lemmatização comparado a outra ferramenta Open-Source já existente e obteve resultados melhores em praticamente todas as línguas analisadas com exceção de persa, estoniano, inglês e húngaro, onde uma possível justificativa foi a quantidade de dados utilizada no treinamento. **(Kestemont et al., 2017)**, por sua vez, propôs utilizar modelo baseado em redes neurais que independente do idioma obteve bons resultados em muitas bases de dados mesmo com ajuste de um único parâmetro do algoritmo, levando a crer que é uma área interessante a ser estudada.

Ainda nesse contexto, **(Lagus e Klami, 2021)** discute em seu trabalho que o processo de lemmatization geralmente se baseia em gramática a partir da morfologia dos idiomas, sendo assim dependente da língua onde está sendo aplicado. Em contrapartida, propõe-se um algoritmo que encontra as palavras normalizadas nas palavras criptografadas matematicamente que são usadas nos modelos de aprendizagem de máquina, sendo assim uma técnica que poderia ser treinada em qualquer idioma, pois atuaria não nas palavras, mas nos vetores já codificados e, de facto, foi comprovado que essa essa metodologia conseguiu resultados superiores ao método tradicional na tarefa de similaridade de documentos.

2.5. AVALIAÇÃO DE PERFORMANCE

Nesse tópico serão definidas as métricas mais comumente utilizadas na medição dos algoritmos de aprendizagem de máquina. Para demonstrar de forma mais clara, será levada em consideração um problema de classificação binária, onde *Tp* significa verdadeiro positivo, Tn corresponde a verdadeiro negativo, Fp falso positivo e Fn significa falso negativo.

2.5.1. Acurácia

Segundo **(Chicco e Jurman, 2020)**, muitos pesquisadores acreditam que a métrica de desempenho mais razoável é a acurácia. A acurácia corresponde à razão entre a quantidade de acertos na classe dividida pela quantidade de amostras e, devido a isto, não há impacto no caso de problemas com mais de duas classes. Ou seja, a acurácia é responsável por demonstrar a efetividade do algoritmo em acertar as classes. A equação 1 representa essa métrica.

Equação 1 - Acurácia

$$acurácia = \frac{Tp + Tn}{Tp + Tn + Fp + Fn}$$

Apesar de ser uma métrica bastante difundida, ela permite uma visão errônea, caso a base de dados não esteja balanceada. Em contextos médicos, na predição de uma enfermidade por exemplo, a frequência de uma das classes tende a ser bem mais alta que a outra, isso faz com que um modelo não-genérico que sempre chuta na classe com maior taxa de ocorrência gerará um valor praticamente perfeito de acurácia.

2.5.2. Precisão

A métrica de precisão indica o poder preditivo do algoritmo ao calcular a percentagem associada à classe positiva ou negativa escolhida para o cálculo **(Sokolova et al., 2006)**. Em outras palavras, caso queira descobrir a precisão positiva, é calculada a percentagem de

previsões verdadeiras que foram realizadas pelo modelo e estavam corretas. A equação 2 representa essa métrica.

Equação 2 - Precisão

$$precisão = \frac{Tp}{Tp + Fp}$$

Buscar uma alta precisão geralmente está vinculada com o objetivo de evitar erros ao identificar uma classe. Por exemplo, se eu quero prever os clientes que meu banco deve ofertar crédito e que irão pagar, assim eu irei obter retorno. No entanto, seria um problema entregar essa oferta para quem não poderá pagar. Logo, se eu identificar um cliente como positivo erroneamente (para oferta de crédito), este erro gerará prejuízos para o banco.

2.5.3. Recall

A métrica de recall indica a probabilidade da minha previsão positiva ser verdadeira **(Sokolova et al., 2006)**. Nesse caso, ao invés de dividir os dados baseados na previsão, divide-se os dados baseados nas classes reais, verificando se existem dados que eram da classe positiva, mas que o modelo falhou em encontrá-los. A equação 3 representa essa métrica.

Equação 3 - Recall

$$recall = \frac{Tp}{Tp + Fn}$$

Em alguns contextos, a métrica recall com níveis baixos está vinculada a erros graves, como no diagnóstico de doenças crônicas. Por exemplo, levando que a classe positiva fosse referente ao diagnóstico de cancro de pulmão, o recall leva em conta quantas vezes deixou-se passar casos positivos, ou seja, informando que não há cancro quando na verdade há.

2.5.4. Medida F (F-Measure)

A medida F, ou F-Measure, é uma métrica composta que tem como parâmetros a precisão e o recall que corresponde a média harmônica de ambas. Quando $\beta = 1$, essa métrica está balanceada e é conhecida como F1-Score. A equação 4 representa essa métrica.

Equação 4 - F1 Score

$$F - measure = \frac{(\beta^2 + 1) * precision * recall}{\beta^2 * precision + recall}$$

Apesar de ser uma métrica bastante utilizada na literatura, existem alguns estudos que contrariam a utilização dessa métrica em todo e qualquer contexto. Em **(Powers, 2016)** aponta uma série de situações onde não se deve usar a medida F: quando o foco está em uma única classe, quando existe viés devido a classe majoritária, quando a distribuição real e a distribuição prevista são idênticas, quando não leva em conta os verdadeiros negativos ou quando fornece diferentes ótimos de outras abordagens.

2.5.5. Área abaixo da Curva ROC

Em **(Bradley, 1997)** é descrito a utilização da área abaixo da curva de Característica de Operação do Recetor, conhecida como curva ROC, como métrica de avaliação de performance para classificadores. A curva ROC é mostrada pela plotagem das taxas de falso positivo e falso negativo em diferentes limiares de classificação, mostrando assim uma relação entre a sensibilidade (recall) do modelo e a especificidade. Pela própria definição, a curva ROC depende da variabilidade no decorrer dos limiares e a área sob a curva seria uma forma de simplificar transformando a medida em um único valor.

Apesar de não gerar grandes diferenças na análise de diferentes modelos nos problemas de classificação, **(Bradley, 1997)** afirma que a área abaixo da curva ROC possui características desejáveis como métrica de performance: aumento de sensitividade nos testes de Análise de Variância **(ANOVA)**, é independente ao limiar de decisão escolhido, mostra quão bem separados são as classes negativas e positivas, é invariante as probabilidades da classe a priori, e entrega uma indicação da efetividade dos algoritmos, uma vez que dá baixas pontuações para algoritmos aleatórios ou que retornam sempre à mesma classe.

3. AREA DE APLICAÇÃO: RECURSOS HUMANOS

3.1. INTELIGÊNCIA ARTIFICIAL E SUA APLICAÇÃO

A maioria dos algoritmos de inteligência artificial já existem a algum tempo. O conceito de aprendizado profundo vem desde 1943 quando Walter Pitts e Warren McCulloch criaram um algoritmo que simulava a ligação dos neurônios da mente humana. Até então o uso desta tecnologia era limitado pela qualidade dos computadores da época.

Com o avanço na produção de peças para computador e o aumento do poder computacional, a aplicabilidade dos algoritmos de inteligência artificial cresceram exponencialmente e várias áreas tem utilizado algoritmos para melhorar seus produtos e serviços.

Dentro muitas aplicações para a inteligência artificial, algumas das comumente utilizadas são:

- **Chatbots:** A criação de algoritmos que conduzem por meio de inteligência artificial conversas online. Normalmente é utilizado no serviço de atendimento ao cliente.
- **Assistentes Pessoais:** Grandes empresas de tecnologia, como Amazon, Apple e Google possuem assistentes pessoais que utilizam do reconhecimento de voz para auxiliar as pessoas em tarefas do dia a dia, como ler uma mensagem, buscar em ferramentas de pesquisas e até mesmo controlar casa inteligentes.
- **Recomendações de produtos e Serviços:** Sites de vendas como Amazon e FNAC e prestadores de serviços de transmissão online como Netflix e HBO utilizam inteligência artificial para definir perfis de usuários e assim melhorar a recomendação de produtos e serviços.

3.2. INTELIGÊNCIA ARTIFICIAL APLICADA NO MERCADO DE TRABALHO

A área de Recursos Humanos vem aproveitando o crescente uso da inteligência artificial para melhorar os processos de contratação e manutenção de colaboradores. Em um processo de recrutamento atual é comum a solicitação de um preenchimento de dados além do envio do curriculum vitae. Estes preenchimentos de informações podem ir de um simples preenchimento de habilidade até um teste psicológico.

Com o acesso a estes dados e o pré treinamento de um algoritmo, as empresas de recursos humanos se habilitam a selecionar candidatos que melhor se enquadrem no perfil procurado, este sendo o uso mais comum da inteligência artificial no setor.

Dentre outras utilizações pode-se destacar também:

- Melhor entendimento dos colaboradores: Empresas como a Neobrain utilizam de inteligência artificial para auxiliar no melhor entendimento dos colaboradores da empresa. Dentre outros, um benefício que se destaca é a diminuição da rotatividade de funcionários.
- Treinamento de colaboradores: A utilização de inteligência artificial para recomendar cursos e treinamento para que os colaboradores mantenham seus conhecimentos atualizados, atinja um próximo passo em sua carreira ou até mesmo transacione sua carreira para outra área dentro da própria empresa.

4. METODOLOGIA

Como já referenciado anteriormente, este trabalho busca escolher a melhor técnica de extração de habilidades de vagas de emprego. Dito isto, o foco está nas habilidades técnicas para a área desenvolvimento web e o objetivo é selecionar as técnicas que nos permita além de identificar habilidades já conhecidas, identificar também habilidades emergentes que possam aparecer ao longo dos anos para esta mesma área de trabalho.

Esta pesquisa é dividida em três etapas.

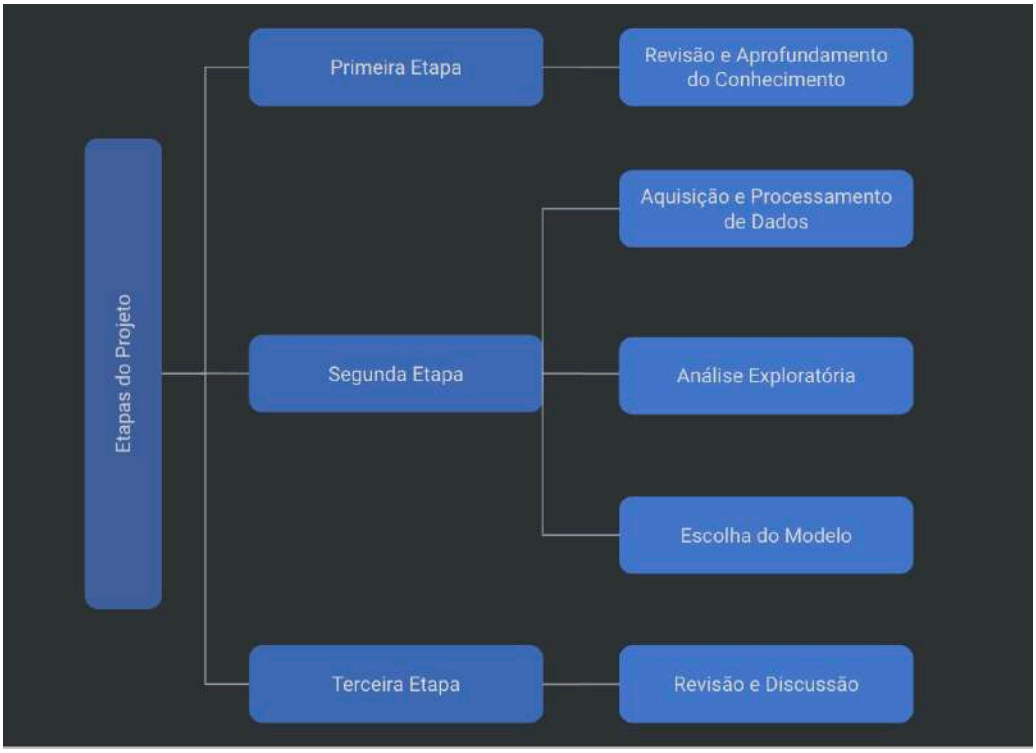
Na primeira etapa, foi feito toda a revisão de conhecimento necessário acerca das técnicas disponíveis para extração de habilidades em texto, à fim de aprofundar o conhecimento na área.

Na segunda, que é mais bem explicada no próximo tópico, será feita a aquisição e processamento dos dados, análise exploratória dos dados, escolha e criação do modelo de extração de habilidades.

Na última etapa o levantamento de resultados será apresentado assim como as discussões a respeito do mesmo e sugestões para trabalhos futuros.

A Figura 2 a seguir materializa a metodologia a ser utilizada:

Figura 2 - Etapas do Projeto



(Fonte: Próprio Autor)

5. ETAPA DE DESENVOLVIMENTO

5.1. AQUISIÇÃO DE DADOS

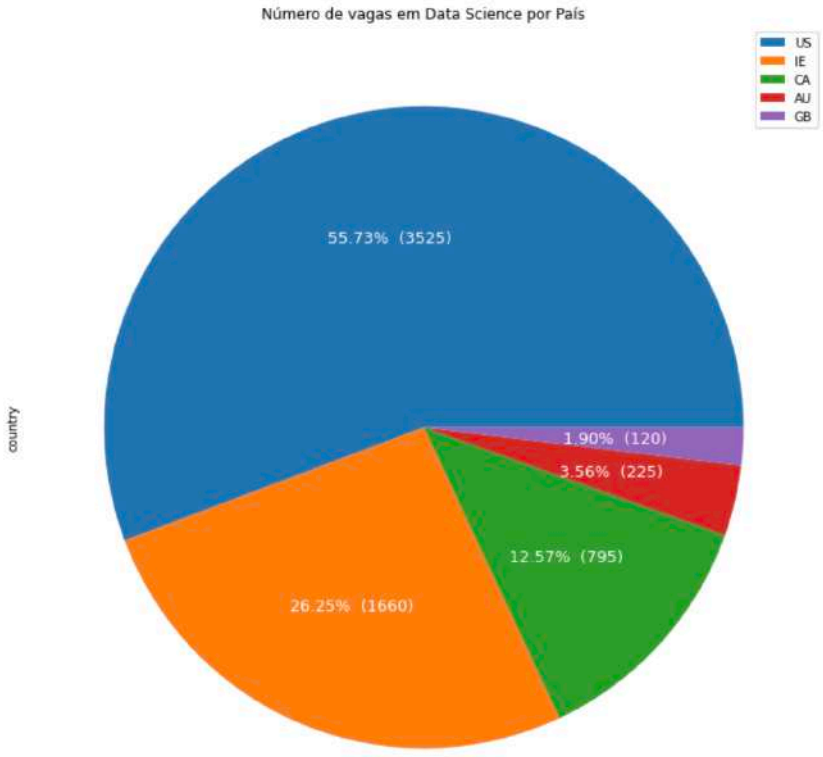
A criação do conjunto dados utilizado neste projeto foi feita por meio da coleta do site Indeed utilizando uma lista de vagas de trabalho de países nativos da língua inglesa sendo eles Austrália, Canadá, Estados Unidos, Grã-Bretanha e Irlanda.

Para criação do conjunto de dados inicial foram buscadas vagas de emprego na área de Data Science utilizando as palavras-chave *"Data Science"* no site *Indeed* nas 5 maiores cidades por população, de cada um dos países citados.

Para coleta de forma automatizada foi criado um *web scraper* utilizando a biblioteca Selenium.

Os dados foram salvos em pastas de acordo com o seu respetivo país em formato Excel (.XLSX), e somados resultaram em um conjunto de dados inicial com 6325 linhas como mostra a Figura 3.

Figura 3 - Distribuição das Vagas de Emprego



Fonte: Próprio Autor

5.2. PRÉ PROCESSAMENTO DE DADOS

Como já descrito anteriormente a etapa de pré processamento de dados visa lapidar os dados convertendo-os de dados brutos para dados preparados. Está é uma fase essencial que precede o treinamento do modelo, garantindo que os dados que serão injetados no modelo são confiáveis.

5.2.1. Seleção de Recursos (Feature Selection)

Quando se fala sobre análise de textos, cada palavra dentro do corpo do texto é considerada um recurso (feature). Neste projeto a etapa de feature selection será dividida em duas, já que primeiro é preciso selecionar quais colunas do conjunto de dados serão utilizadas e quais não, e em um segundo momento selecionar e limpar os dados da descrição de emprego.

5.2.1.1. Seleção de Dados

A primeira etapa da limpeza do conjunto de dados se dá pela seleção dos recursos (features) que serão utilizados e da eliminação dos demais. A coluna *description* contém o texto de cada vaga de emprego, e *job_title* por sua vez a informação sobre o título da vaga. O título da vaga será utilizado para garantir que as vagas a que estamos buscando habilidades correspondem com a área de Data Science.

Mantem-se então 2 das colunas do conjunto de dados que serão utilizadas, *description* e *jobtitle.* As demais colunas são descartadas.

Como última etapa da seleção de dados, foi preciso comparar os títulos das vagas encontradas pelo *web scraper,* com o termo utilizado para a busca no site de empregos Indeed. Isto foi feito para garantir que as vagas de emprego que formam o conjunto de dados são relevantes para a área em questão. Os títulos foram comparados utilizando a similaridade do cosseno. Esta similaridade retorna um valor decimal entre 0 e 1 que diz quão similar dois vetores são, quanto maior o valor, maior a similaridade entre os textos. Para a execução deste projeto foi aplicado um filtro selecionando as 40 vagas com maior similaridade entre o título de emprego e o termo buscado: “Data Science”. Todas as 40 vagas selecionadas tiveram similaridade 1, o que quer dizer que o título da vaga corresponde exatamente ao título buscado.

5.2.1.2. Limpeza do texto

Os dados buscados pelo web scraper vieram em formato HTML então o primeiro passo da limpeza do texto foi remover as *tags* HTML.

Após análise gramatical na estrutura das vagas de emprego, nota-se que a maioria das habilidades são substantivos (*Nouns*). Antes de utilizar técnicas para limpeza do texto como a remoção de palavras irrelevantes (*stopwords*) foi criado uma nova coluna chamada *nouns* no conjunto de dados como uma lista, que contém todas as frases que contém substantivos para a respetiva vaga de emprego. Foi feito uma cópia da coluna *description* que não será processada com a remoção de palavras irrelevantes.

Como já mencionado a remoção de palavras irrelevantes (*stopwords*) é parte crucial quando se trabalha com textos. É nesta etapa que palavras que não carregam informações são removidas. Para a remoção de stopwords foi utilizada a biblioteca Spacy.

5.3. ANÁLISE EXPLORATÓRIA

A análise exploratória é o processo de investigação inicial em um projeto de análise de dados e tem como objetivo resumir as características dos dados. Para esta análise exploratória foi utilizada a coluna *description* (que ainda contem as stopwords) para não descaracterizar o texto e ter uma imagem real dos dados.

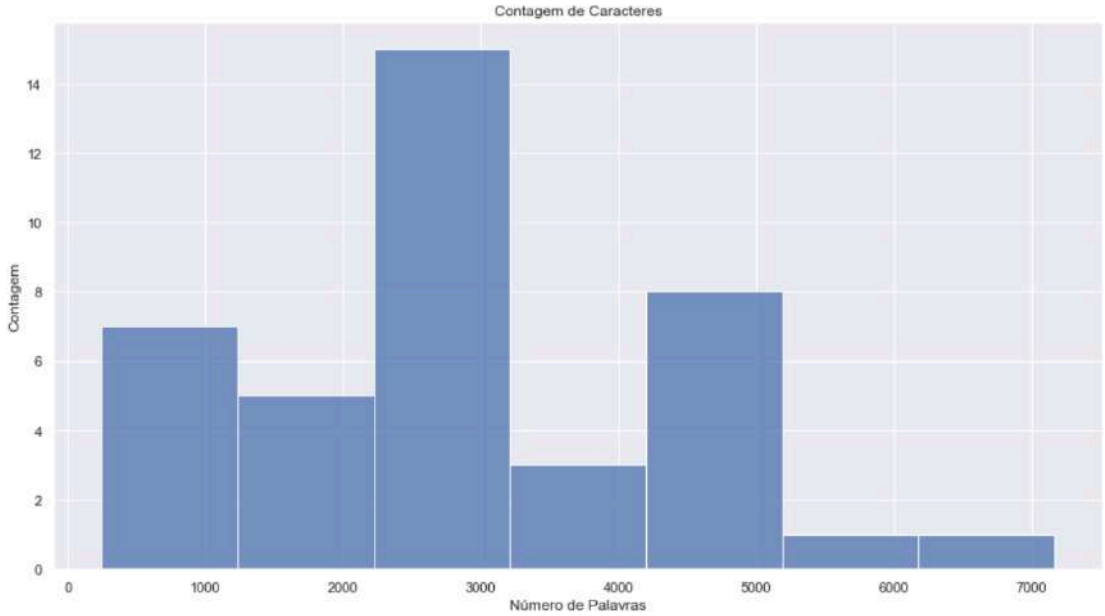
5.3.1. Contagem de Caracteres e Palavras

As figuras 4 e 5 a seguir nos mostram a contagem de caracteres e de palavras respetivamente.

Pode-se inferir que o número de caracteres de cada vaga de emprego está entre 200 e 7200 e que a maioria dos documentos tem entre 2000 e 3000 caracteres.

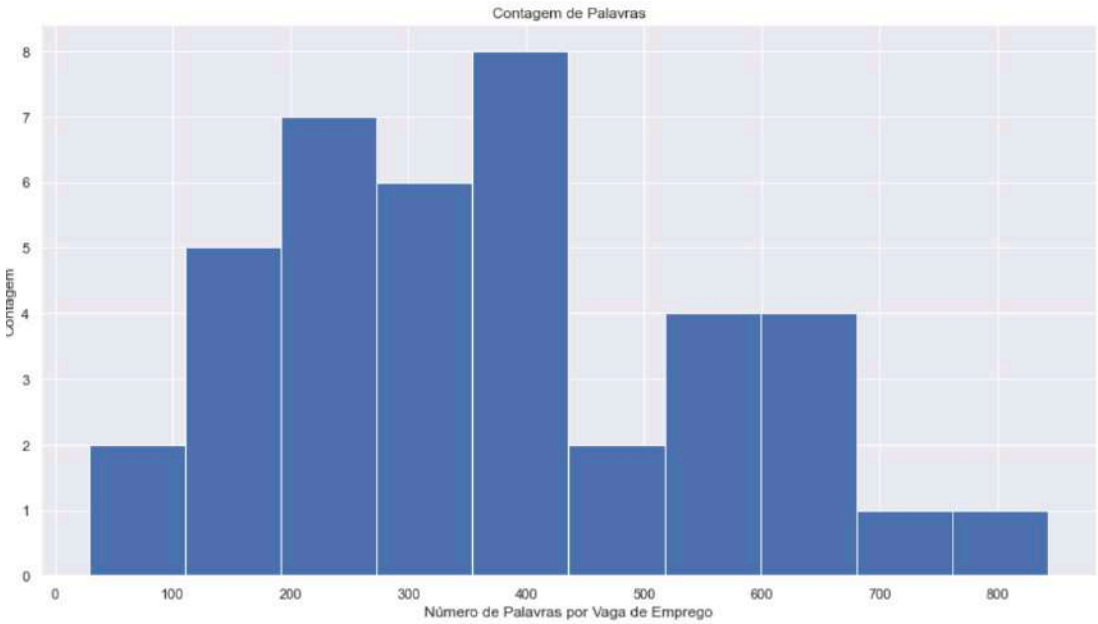
Quanto contagem de palavras o número fica entre 20 e 900 sendo 200 a 400 palavras o mais comum.

Figura 4 - Contagem de Palavras



Fonte: Próprio Autor

Figura 5 - Contagem de Palavras

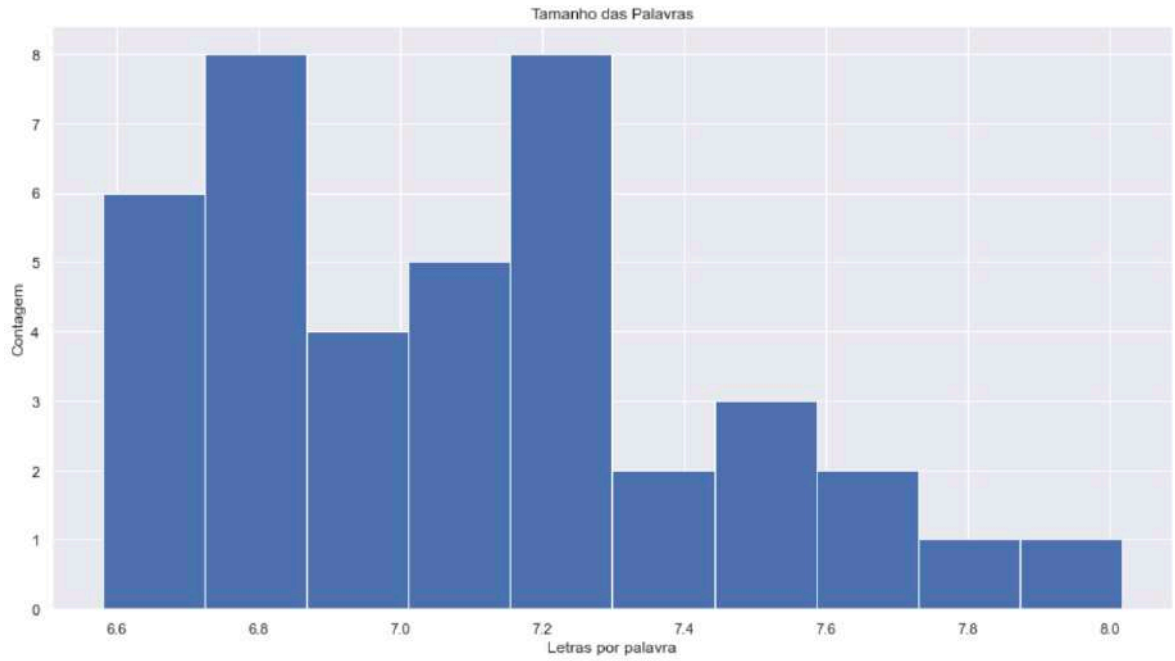


Fonte: Próprio Autor

5.3.2. Comprimento das Palavras

O comprimento médio da palavra varia entre 6.6 a 8, sendo 6.8 e 7.2 com 8 palavras cada, os comprimentos mais comuns como mostra a Figura 6.

Figura 6 - Tamanho das Palavras

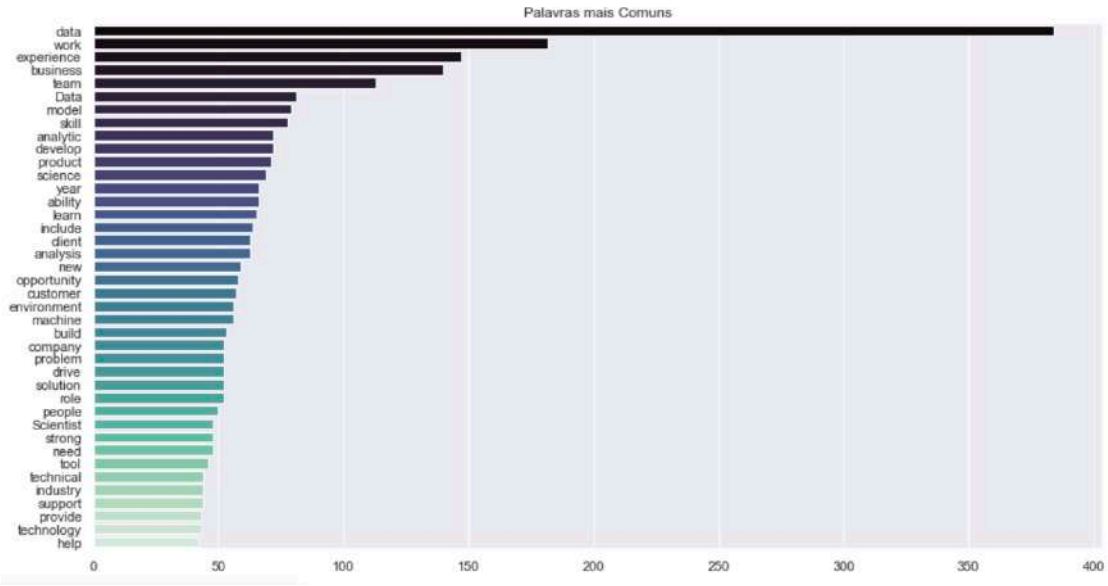


Fonte: Próprio Autor

5.3.3. Palavras Comuns

Na Figura 7 pode-se ver que *data* (dados) aparece como a palavra mais comum do corpus, aparecendo quase quatrocentas vezes, seguida por *work* e *experience*. Podemos ver também que as palavras *learn* (aprendizado*)* e *machine* (máquina) figuram entre as 30 palavras que mais aparecem, apontando em um primeiro momento que talvez *Machine Learning* (aprendizado de máquina) seja uma das habilidades mais requisitadas para esta área.

Figura 7 - Palavras Comuns

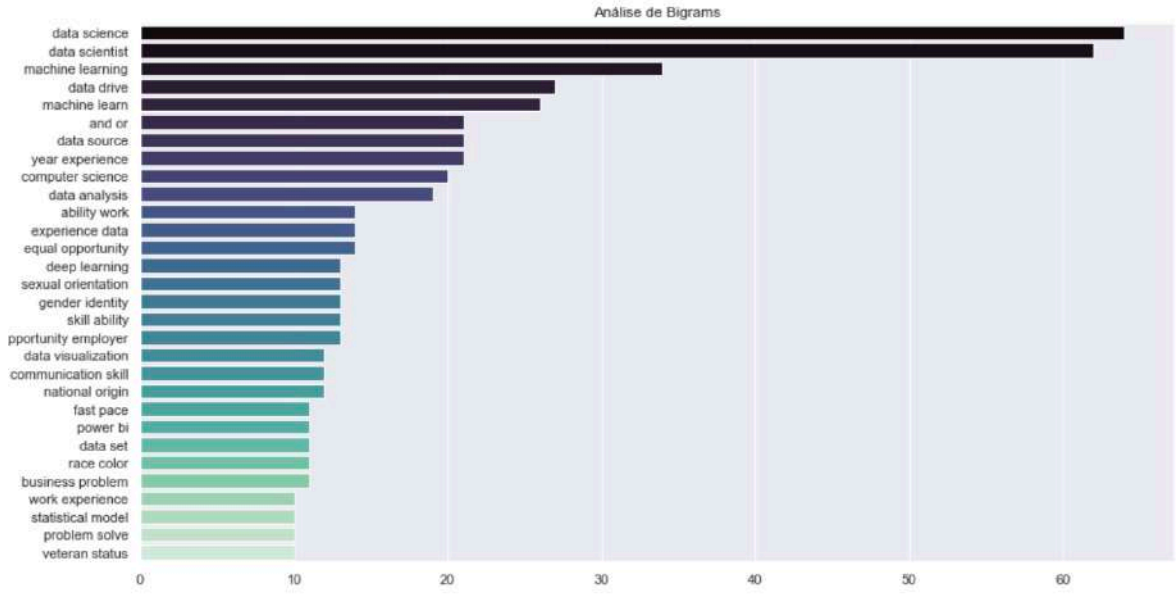


Fonte: Próprio Autor

5.3.4. Palavras em Par

A Figura 8 mostra a contagem de palavras em par (*bigrams*). O resultado mostrou como *Data Science* sendo o par de palavras que mais aparece, seguido por *Data Scientist* e *Machine Learning*, apontando uma segunda vez que ML pode ser uma das habilidades que mais aparecem em vagas de emprego para a área de ciência de dados.

Figura 8 - Paravras em Par



Fonte: Próprio Autor

5.4. ESCOLHA DA TÉCNICA DE EXTRAÇÃO DE HABILIDADES

Foram selecionadas três das técnicas como potenciais candidatos na extração de habilidades sendo elas: TF-IDF, Word Embeddings e Named Entity Recognition.
Nos subtópicos seguintes será brevemente explicado o porque cada algoritmo pode ser útil na extração de habilidades, depois serão descritos os pontos positivos e negativos sobre cada uma destas técnicas, e por fim uma destas técnicas será selecionada para criação do modelo.

5.4.1. Term Frequency - Inverse Document Frequency (TF-IDF)

Como apresentado anteriormente o TF-IDF é algoritmo que quantifica palavras em uma série de documentos. Os pesos de cada palavras são computados de acordo com a importância desta palavra para o documento e o corpus, quanto mais aparecer no documento maior a importância e quanto menos aparecer no corpus maior importância a palavra passa a ter.
Relativamente ao problema de identificação de habilidades em ofertas de emprego, o TF-IDF se mostra interessante por ser um algoritmo não supervisionado que não necessita de pré-treinamento e com isto vem a poder auxiliar na deteção de habilidades emergentes.
Em contrapartida uma dificuldade que o TF-IDF pode vir a enfrentar na deteção de habilidades em vagas de emprego, é que na maioria das vagas, as habilidades são mencionadas apenas uma vez, diminuindo assim seu peso perante outras palavras que costumam sempre aparecer em vagas de emprego e não são habilidades, como por exemplo as palavras experiência e o próprio título da vaga *Data Scientist.*

5.4.2. Classificador de Substantivos

Para extrair habilidades de um texto por meio de classificação é necessário primeiro escolher qual foco no texto a se classificar. As opções mais coerentes são a classificação de parágrafos ou de frases, que contem habilidades. A primeira opção à primeira vista não parece viável visto que um mesmo parágrafo pode conter mais de uma habilidade, ou até mesmo outro substantivo, dificultando assim a rotulação. Já classificação de frases se mostra mais concreta, visto que raramente dois substantivos irão aparecer no mesmo trecho.
A seguir temos o exemplo de um possível processo de criação inicial com classificadores e a Figura 9 que ilustra o mesmo processo:

- Pré-processamento de dados;
- Extrai-se as frases com substantivos;

- Mapeia-se as frases entre frases com habilidades ou frase sem habilidades;
- Um classificador é escolhido e treinado para prever frases com habilidades. Uma rede neural que utiliza camadas de *Long short-term Memory* pode ser uma potencial candidata;
- Treina-se o modelo escolhido com os dados mapeados;
- Extrai-se os substantivos (habilidades) de novos dados, com as frases que foram previstas como contendo habilidades.

Figura 9 - Processo de Classificação de Habilidades

Fonte: Próprio Autor

A extração por classificador à primeira vista mostra-se promissora na deteção de habilidades em vagas de emprego. Um ponto inicial que pode impactar nos resultados é que o algoritmo de classificação não leva em consideração a sintaxe do texto, apenas selecionando substantivos de frases e os classificando como habilidade ou não.

5.4.3. Reconhecimento de Entidades Nomeadas

O reconhecimento de entidades nomeadas (NER) desempenha um papel importante ao fazer as máquinas entenderem texto escrito por humanos. Ao classificar as entidades contidas dentro de um texto desestruturado, importantes informações, como as habilidades em uma vaga de emprego, podem ser encontradas.

Para o reconhecimento da entidade o algoritmo faz uso de um input *Word Embeddings* e utiliza uma rede neural artificial (comumente CNN ou LSTM) para o entendimento das relações entre palavras, assim conseguindo prever se a palavra pertence a uma entidade ou não.

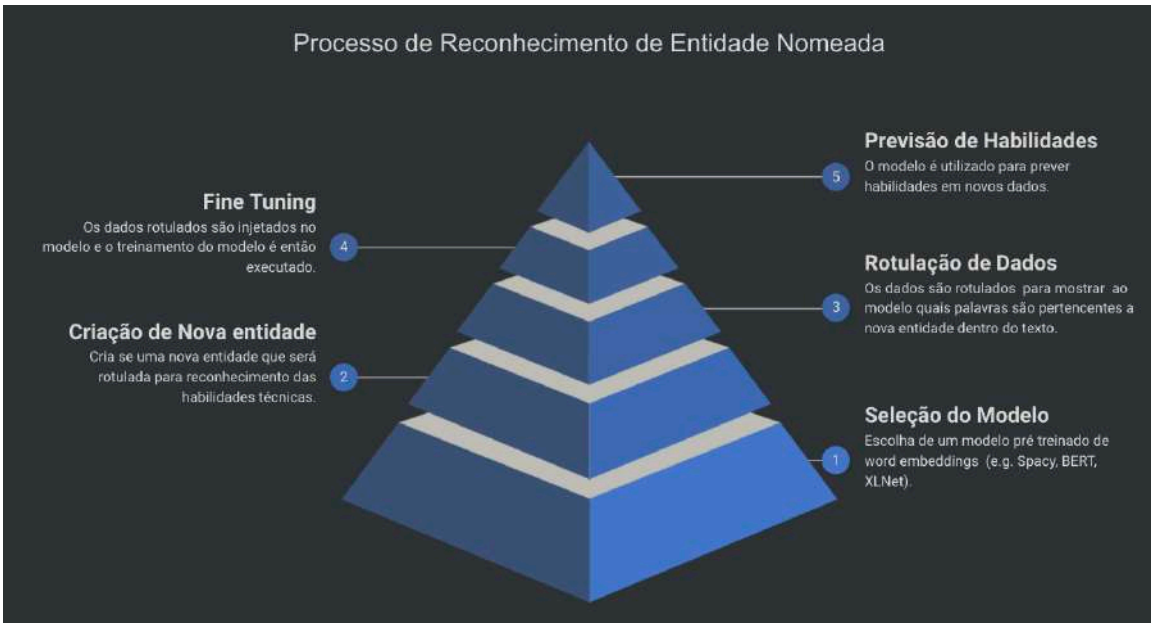
Como já mencionado anteriormente o NER por meio da sintaxe, procura significado entre o relacionamento de palavras, frases e textos e por este motivo a remoção de palavras irrelevantes pode não ser interessante, já que estas palavras podem ser utilizadas pelo modelo para melhor entender a sintaxe do texto.

Uma técnica atual que vem dando bons resultados na área é a aplicação de transferência de aprendizado mais conhecida como *Transfer Learning*. No *Transfer Learning* um modelo pré treinado para reconhecer entidades comuns como pessoas, datas e organizações é utilizado e um ajuste fino é feito para resolver uma atividade específica, como por exemplo o reconhecimento de habilidades em vagas de emprego.

A seguir o processo reconhecimento de entidades e a Figura 10 que demonstra o processo:

- Escolhe-se um modelo de pré-treinado;
- Cria-se uma nova entidade que será treinada;
- Rotula-se os dados apontando ao modelo quais palavras são pertencentes a nova entidade;
- É feito o ajuste fino, ou *Fine Tuning,* do modelo a fim de treinar o modelo para reconhecer as habilidades;
- Extrai-se as entidades (habilidades) de novas vagas de emprego.

Figura 10 – Processo de Reconhecimento de Entidade Nomeada (NER)



Fonte: Próprio Autor

O NER se mostra um potencial candidato na extração de habilidades por utilizar semântica para definir as entidades.
Os pontos negativos do NER são o poder computacional necessário para rodar o algoritmo e a customização que deve ser feita para o *fine tuning*, aonde em cada algoritmo é diferente.

5.4.4. Escolha da Técnica

Após a avaliação cuidadosa feita nos tópicos anteriores sobre os possíveis modelos a serem utilizados, o *Named Entity Recognition* se mostrou uma escolha mais promissora para a extração de habilidades em vagas de emprego. Esta opção se deu ao fato de o NER utilizar modelos robustos pré treinados, que são o estado da arte atual, para entender a proximidade entre palavras e utilizar a sintaxe textual para reconhecimento das entidades.

5.5. MODELOS

Com o pré processamento de dados concluído e a técnica de extração selecionada, os dados foram particionados em dados de treinamento e dados de validação seguindo as proporções de 75% para treinamento e 25% para validação. Estes valores foram escolhidos porque figuram entre o normal aplicado de Machine Learning que costuma ser entre 70% - 30% e 80% - 20% respetivamente para dados de treino e validação.
O estado da arte na área de NER hoje é constituído pela utilização de modelos pré treinados em milhões de dados para tarefas generalizadas, e então um ajuste fino é feito sobre este modelo para resolver uma tarefa específica. Esta técnica é comumente chamada de *Transfer Learning*. Três modelos pré treinados foram selecionados para aplicação do *Transfer Learning*: Spacy, BERT e XLNet.
Spacy é uma biblioteca relativamente nova da área de Natural Language Processing que oferece ferramentas com soluções na área de Natural Language Processing. Uma delas é um modelo de NER que utiliza Word Embeddings e uma Rede Neural Convolucional (CNN) para treinar suas entidades e que tem tido ótimos resultados.
BERT, como já visto anteriormente, é um algoritmo que foi desenvolvido por engenheiros do Google e que utiliza técnicas de aprendizado bidirecional e encoding para resolver tarefas de processamento de linguagem natural por meio de um transformer. A base de aprendizado do BERT envolve esconder certas palavras do texto durante o seu processo de treinamento a fim de tentar prever a palavra correta. Utilizaremos a versão RoBERTa que como já visto é uma variação do BERT, porém mais robusto.

XLNet é modelo auto regressor que também utiliza técnicas bidirecionais. O XLNet utiliza uma técnica chamada Modelagem de Linguagem de Permutação onde no lugar de esconder uma das palavras e tentar prevê-la, como no BERT, permuta as palavras de uma frase de lugar a fim de prever uma das palavras.

Os três modelos foram treinados a título de comparação de performance e no capítulo seguinte, será exposto o resultado de cada modelo para a tarefa proposta neste trabalho.

Para criação da nova entidade e rotulação dos dados algumas considerações foram feitas na definição de o que é uma habilidade técnica ou *hard skill*. Para este trabalho foram consideradas habilidades técnicas:

- Linguagens de programação como Python e R;
- Áreas de conhecimento como *Machine Learning* e *Deep Learning*;
- Bibliotecas e frameworks como *Pandas*, *Numpy* e *Scikit-learn*;
- Aplicativos ou programas como Excel, PowerPoint e Jupyter-Notebook;
- Formações acadêmicas como Ciência da Computação, Matemática, Estatística, Física e Biologia.

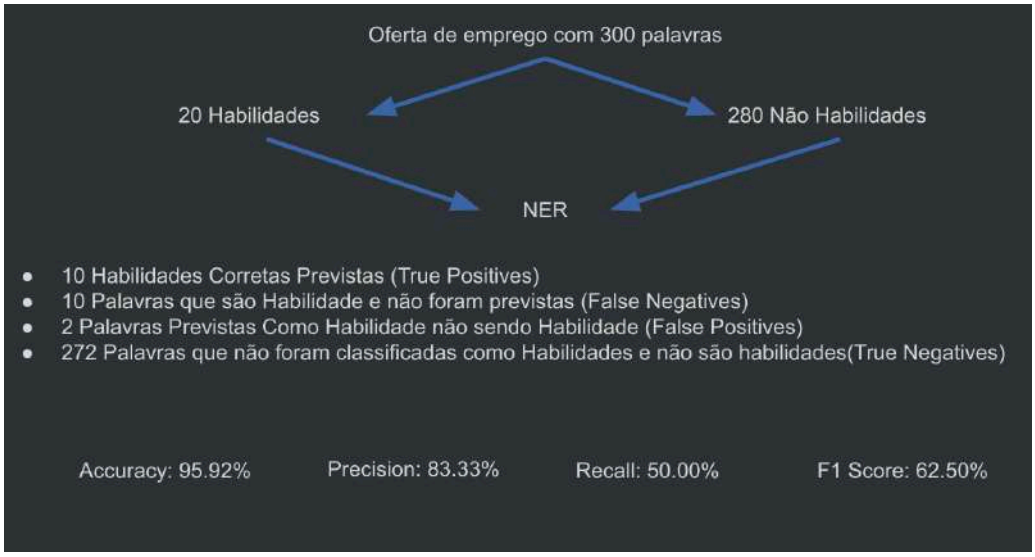
O próximo passo foi o ajuste fino dos modelos, onde cada um deles foi alimentado com os mesmos dados rotulados com o objetivo de treinar os três modelos na tarefa de reconhecimento de habilidades.

Para o treinamento dos modelos a plataforma *Google Colab* foi utilizada na sua versão profissional. Dois dos modelos tem arquiteturas que utilizam Transformers e para aceleração do processo de treinamento foi necessário um processador gráfico integrado ou GPU.

6. RESULTADOS E DISCUSSÃO

Ao utilizar métricas de avaliação para problemas de reconhecimento de entidades, tem-se de levar em consideração que as palavras que não forem classificadas pelo algoritmo e que não forem entidades serão consideradas como negativo verdadeiro. Isto pode levar a um problema em métricas de avaliação que utilizem os números negativos verdadeiros, como por exemplo a acurácia. A Figura 11 ilustra o problema da utilização da acurácia em problemas de NER.

Figura 11 - Exemplo de Métricas



Fonte: Próprio Autor

Foram então utilizadas três métricas para medir a performance do modelo: *Precision*, *Recall* e *F1 Score*.

Para computar as métricas, os seguintes resultados foram analisados:

1. Palavras que foram rotuladas como Habilidades e que também foram previstas pelo algoritmo como Habilidades (*True Positives*);
2. Palavras que não foram rotuladas como habilidades, mas que foram previstas como sendo habilidades pelo algoritmo de NER (*False Positives*);
3. Palavras que foram rotuladas como habilidades, mas que não foram consideradas como habilidades pelo algoritmo de NER (*False Negatives*).

A tabela 1 a seguir mostra o resultado individual de cada um dos três algoritmos de NER para cada documento separado para teste na coluna NER. Os documentos estão enumerados de 1 a

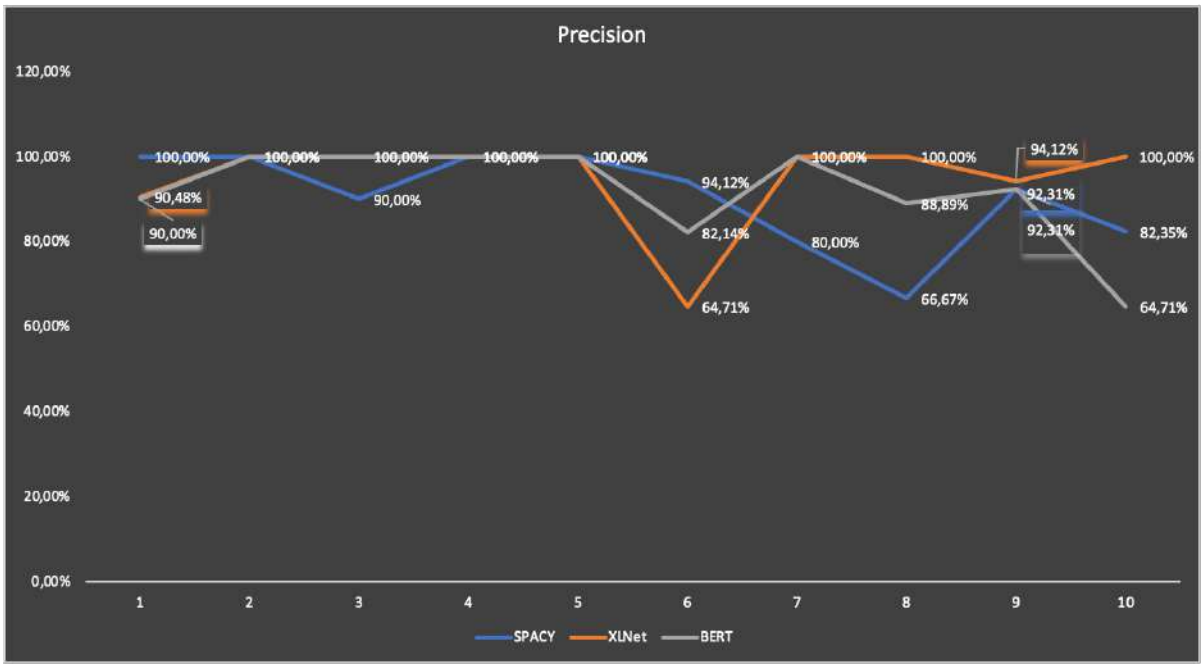
10, e os resultados de cada algoritmo para este documento pode ser encontrado na linha do documento.

Tabela 1 - Resultados dos Algoritmos de NER

NER	Document	Total Habilities	Total-Predicted	Right-Predicions (True Positive)	Wrong-Predictions (False Positive)	Existing skill not found by algo (False Negatives)	Precision	Recall	F1 Score
SPACY	1	29	24	24	0	5	100,00%	82,76%	90,57%
SPACY	2	27	26	26	0	1	100,00%	96,30%	98,11%
SPACY	3	11	10	9	1	2	90,00%	81,82%	85,71%
SPACY	4	16	13	13	0	3	100,00%	81,25%	89,66%
SPACY	5	5	5	5	0	0	100,00%	100,00%	100,00%
SPACY	6	23	17	16	1	7	94,12%	69,57%	80,00%
SPACY	7	12	10	8	2	4	80,00%	66,67%	72,73%
SPACY	8	9	6	4	2	5	66,67%	44,44%	53,33%
SPACY	9	24	26	24	2	0	92,31%	100,00%	96,00%
SPACY	10	14	17	14	3	0	82,35%	100,00%	90,32%
XLNet	1	29	21	19	2	10	90,48%	65,52%	76,00%
XLNet	2	27	20	20	0	7	100,00%	74,07%	85,11%
XLNet	3	11	9	9	0	2	100,00%	81,82%	90,00%
XLNet	4	16	15	15	0	1	100,00%	93,75%	96,77%
XLNet	5	5	4	4	0	1	100,00%	80,00%	88,89%
XLNet	6	23	17	11	6	12	64,71%	47,83%	55,00%
XLNet	7	12	5	5	0	7	100,00%	41,67%	58,82%
XLNet	8	9	5	5	0	4	100,00%	55,56%	71,43%
XLNet	9	24	17	16	1	8	94,12%	66,67%	78,05%
XLNet	10	14	12	12	0	2	100,00%	85,71%	92,31%
BERT	1	29	30	27	3	2	90,00%	93,10%	91,53%
BERT	2	27	27	27	0	0	100,00%	100,00%	100,00%
BERT	3	11	10	10	0	1	100,00%	90,91%	95,24%
BERT	4	16	16	16	0	0	100,00%	100,00%	100,00%
BERT	5	5	5	5	0	0	100,00%	100,00%	100,00%
BERT	6	23	28	23	5	0	82,14%	100,00%	90,20%
BERT	7	12	10	10	0	2	100,00%	83,33%	90,91%
BERT	8	9	9	8	1	1	88,89%	88,89%	88,89%
BERT	9	24	26	24	2	0	92,31%	100,00%	96,00%
BERT	10	14	17	11	6	3	64,71%	78,57%	70,97%

A Figura 12 a seguir mostra um gráfico de linhas que representa os resultados da métrica *Precision*. Podemos constatar uma melhor performance do algoritmo XLNet que chega a cem por centro de precisão em 7 dos 10 documentos e atinge acima de noventa por cento 9 deles. O algoritmo Bert obteve uma performance estável estando pouco abaixo do XLNet em termos de precisão, enquanto o algoritmo Spacy oscilou de documento para documento.

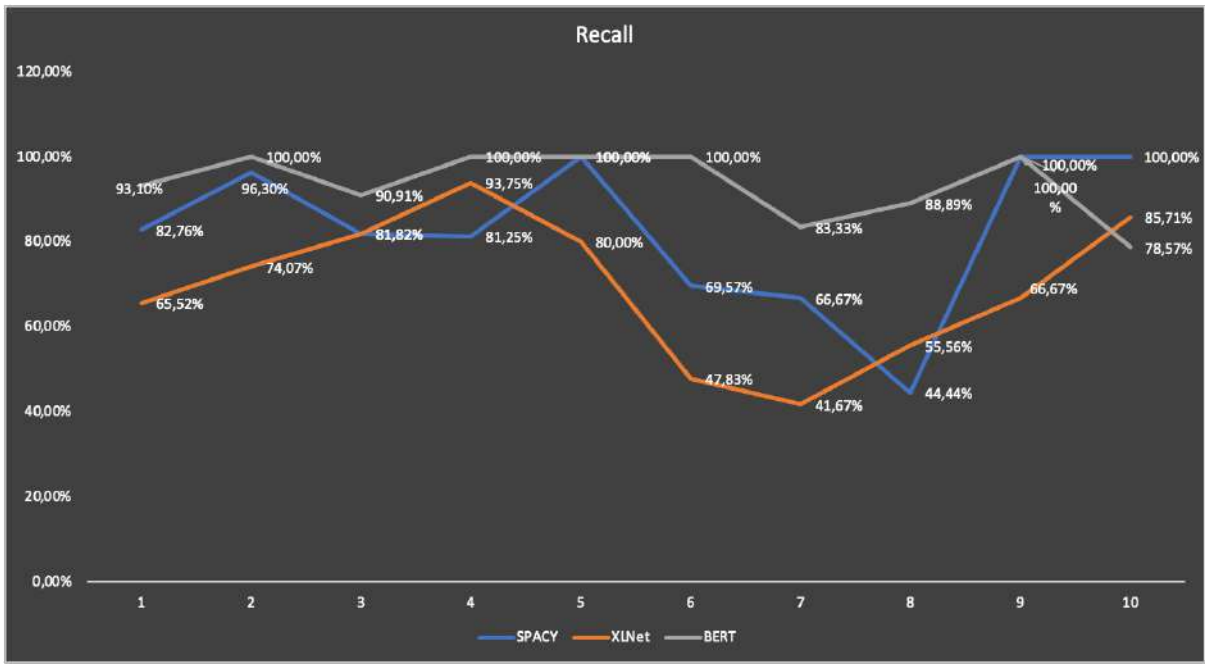
Figura 12 - Precision



Fonte: Próprio Autor

Quanto ao Recall pode-se confirmar na Figura 13 que o BERT foi o algoritmo mais estável e com melhor performance geral, alcançando em 7 documentos resultados acima de noventa por cento, sendo 5 deles cem por cento. Os outros dois algoritmos tiveram uma baixa performance quando comparado ao BERT, em alguns documentos atingindo valores abaixo de cinquenta por cento.

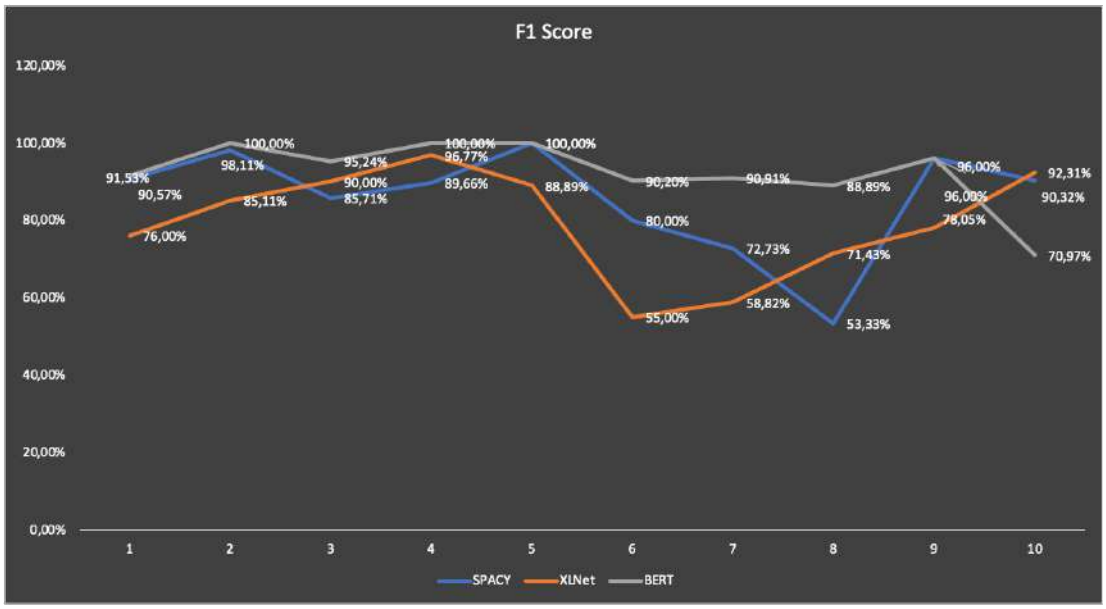
Figura 13 – Recall



Fonte: Próprio Autor

No F1 Score que é uma métrica que procura equilibrar os resultados de Precision e Recall, novamente o BERT se destacou, tendo a melhor performance dos três algoritmos em 9 dos 10 documentos e sendo que em 8 deles o valor do F1 Score foi acima de noventa por cento. Pode-se notar que o mal desempenho no Recall afetou também o resultado do XLNet e do Spacy no F1 Score.

Figura 14 - F1 Score



Fonte: Próprio Autor

Para chegar a um resultado final sobre qual modelo utilizar, foi computado a média aritmética dos resultados de cada modelo para cada documento, em cada uma das métricas.

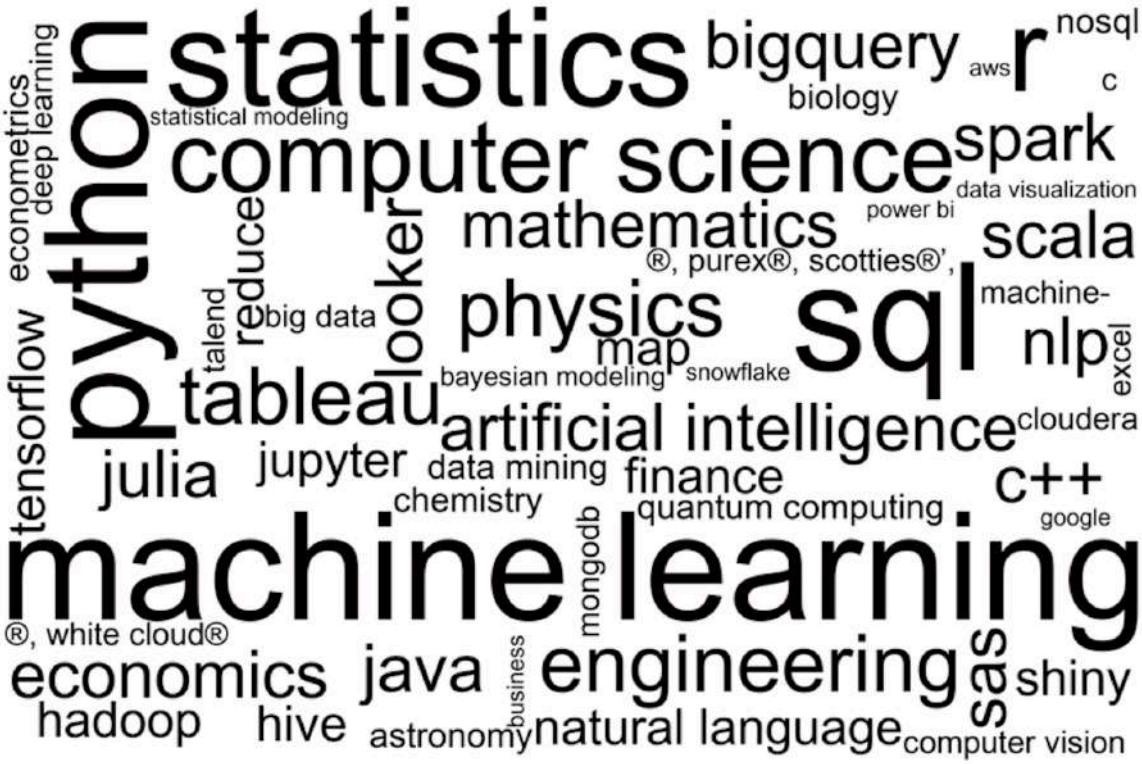
A Tabela 2 mostra a seguir o resultado final, onde comprova-se que em termos de precisão vence o XLNet e nos outros dois quesitos o algoritmo que obteve o melhor resultado foi o BERT. Tem de se notar também que a diferença em termos de precisão entre o XLNet e o BERT foi de 3.13%, no Recall a diferença entre BERT e SPACY foi de 11.2% e no F1 Score a diferença entre BERT e Spacy foi de 6.73%.

Tabela 2 - Resultado Final

NER	Precision	Recall	F1 Score
SPACY	90,54%	82,28%	85,64%
XLNet	94,93%	69,26%	79,24%
BERT	91,80%	93,48%	92,37%

Este resultado mostra que o algoritmo XLNet acertará na maioria das vezes que recomendar habilidades, não recomendando palavras incorretas em quase 95% das vezes, porém mostra também que parte das habilidades presentes no documento acabarão não sendo detetadas por este algoritmo. O único caso para utilização deste algoritmo é se o objetivo do projeto for prever habilidades com a maior precisão sem levar em consideração o total de habilidades. A figura 15 mostra em formato de nuvem de palavras, ou *WordCloud*, o resultado do algoritmo XLNET.

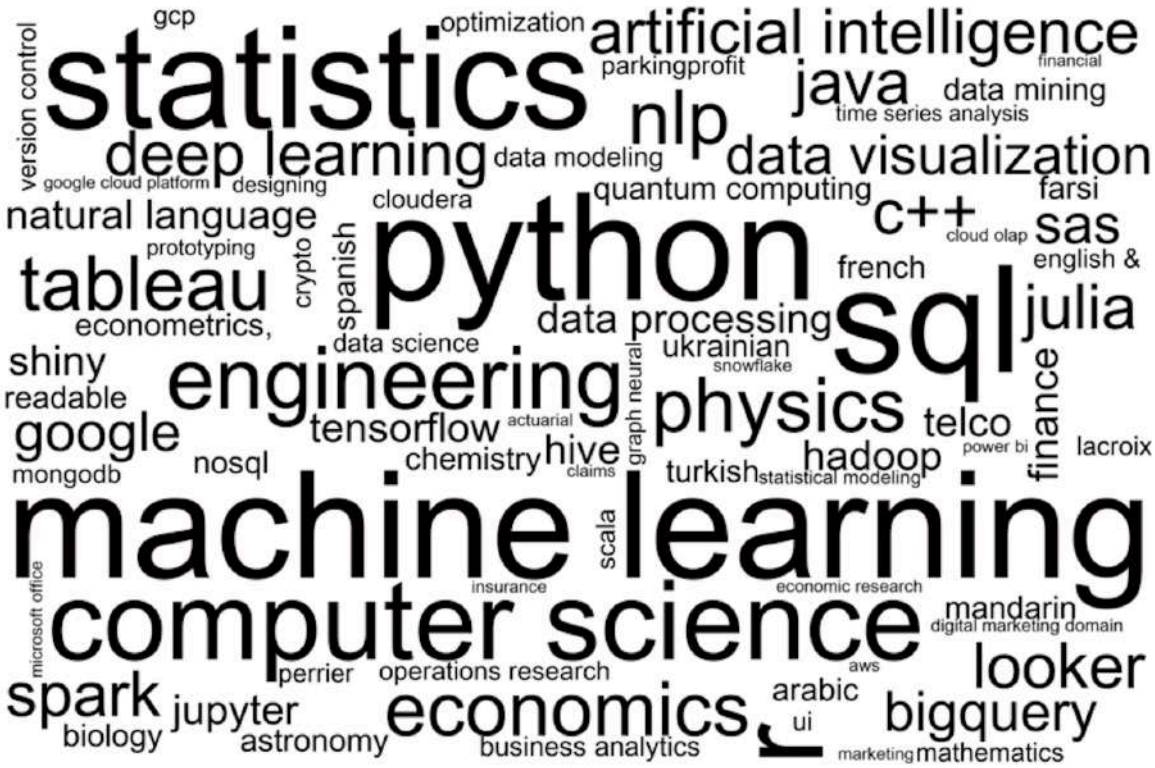
Figura 15 - WordCloud XLNet



Fonte: Próprio Autor

O Spacy teve um bom resultado na precisão, e um resultado mediano no Recall. Por se tratar de um algoritmo leve e de fácil treinamento pode ser uma escolha razoável caso custo de hardware ou tempo de treinamento, sejam pontos relevantes para o projeto. A figura 16 apresenta o resultado do algoritmo Spacy.

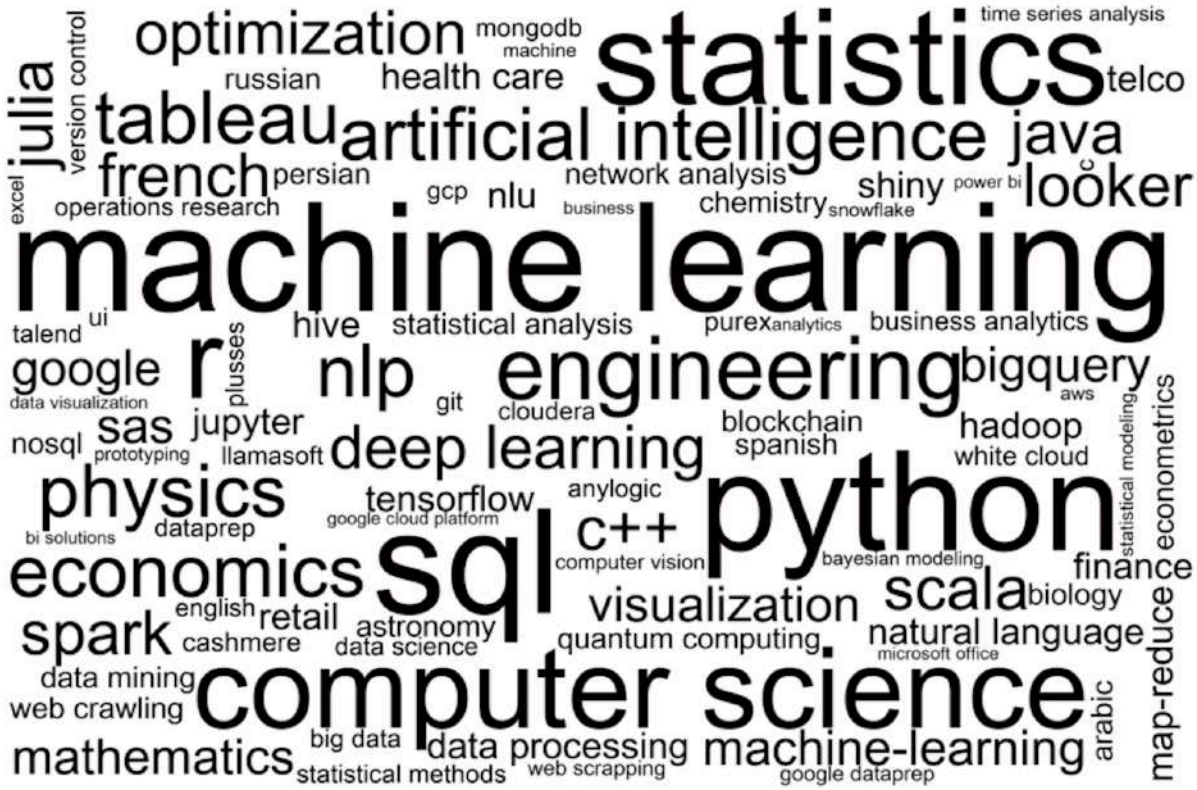
Figura 16 - WordCloud Spacy



Fonte: Próprio Autor

BERT se mostrou um algoritmo estável e com excelentes resultados em todas as métricas, sendo a escolha mais fiável entre os três algoritmos. O ajuste fino no algoritmo se mostrou eficaz, mostrando que para o problema proposto neste trabalho, a escolha da técnica de entidades nomeadas aliada com a transferência de conhecimentos foi uma escolha acertada. A figura 17 apresenta o resultado do algoritmo BERT.

Figura 17 - WordCloud BERT



Fonte: Próprio Autor

7. CONCLUSÕES

Quando se iniciou o trabalho de pesquisa constatou-se que há uma dificuldade, por conta da evolução do mercado de trabalho, no acompanhamento da das habilidades necessárias para vagas de emprego e por isto, mostrou-se importante o desenvolvimento de um algoritmo que detetasse habilidades técnicas em vagas de emprego.

Diante disto, a pesquisa teve como objetivo geral a criação de um modelo de Natural Language Processing para extração de habilidades técnicas na área de ciência de dados. Constata-se então, que o objetivo geral foi atendido porque efetivamente o trabalho obteve sucesso ao utilizar técnicas de NLP e obteve resultados significativos nas métricas de avaliação.

O primeiro objetivo específico deste projeto era analisar as técnicas de processamento de dados e avaliar se, e quais delas se enquadravam para o problema. Mesmo com a não aplicação da remoção de *stopwords* devido a escolha do algoritmo, considera-se este objetivo atendido, visto que, a análise sobre a utilização de cada técnica de processamento foi levada em consideração e chegou-se a conclusão de que a não utilização desta técnica seria a melhor escolha.

O segundo objetivo específico, que era explorar os modelos de aprendizado de máquina a fim de compreender as diferentes técnicas de extração de habilidades de emprego, também foi atingido, ao ser feita uma análise cuidadosa na revisão da literatura sobre cada técnica de aprendizado de máquina supervisionada e não supervisionada, NLP e *Transfer Learning,* para posteriormente selecionar três potenciais candidatos na extração de habilidades e por fim selecionar NER como modelo final.

O último objetivo específico era a aplicação da técnica mais promissora para extração de habilidades, que foi também foi atendido na criação dos três modelos: XLNet, Spacy e BERT.

Uma outra conclusão a ser levada em consideração é que a habilidade *Machine Learning*, que foi apontada na parte de análise exploratória como uma das habilidades que mais apareciam, também foi a habilidade que mais apareceu nos três modelos utilizados neste projeto.

Assim, conclui-se que este projeto obteve sucesso na extração das habilidades para a área de ciência de dados.

8. LIMITAÇÕES E RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Para a realização deste projeto a metodologia apresentada foi primeiro a seleção de uma área inicial para a extração de habilidades, depois foi feito o aprofundamento no tema na revisão bibliográfica seguido de seleção das técnicas de extração, criação do modelo e apresentação de resultados.

Diante da metodologia proposta, esta pesquisa focou em uma área de emprego específica. Uma dificuldade para expansão deste trabalho para outras áreas é o conhecimento a cerca das habilidades de cada área específica. É necessário um profundo conhecimento sobre a área, para que habilidades técnicas não passem despercebidas na rotulação.

Ainda na rotulação de dados, a quantidade de dados rotulados acaba por ser também uma limitação, por conta do tempo necessário para execução desta tarefa. Um número maior de dados poderia ter sido rotulado a fim de buscar uma melhor performance, e assim tendo também um maior número de dados de teste para avaliar a performance do modelo.

Como recomendação para trabalhos futuros, eu recomendo a criação de uma nova entidade para identificação de habilidades interpessoais, ou *soft skills* e a rotulação destas habilidades que posteriormente seriam detetadas pelo modelo.

Outra recomendação importante, como já dito acima, é a expansão do modelo para diferentes áreas de trabalho. Uma possível área a ser explorada inicialmente pode ser a área de tecnologia como um todo.

Por fim, minha última recomendação seria a exploração do modelo de classificação de frases com habilidades proposto na fase de seleção de técnicas de extração, que apesar de não levar em consideração a semântica do texto pode vir a ter bons resultados.

9. BIBLIOGRAFIA

Adam, M., Wessel, M., & Benlian, A. (2021). AI-based chatbots in customer service and their effects on user compliance. Electronic Markets, 31(2), 427-445.

Agerri, R., Bermudez, J., & Rigau, G. (2014, May). IXA pipeline: Efficient and Ready to Use Multilingual NLP tools. In *LREC* (Vol. 2014, pp. 3823-3828).

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, *6*(1), 37-66.

Akhmetov, I., Pak, A., Ualiyeva, I. M., & Gelbukh, A. (2020). Highly language-independent word lemmatization using a machine-learning classifier. *Computación y Sistemas*, *24*(3).

Al Hasan, M., Chaoji, V., Salem, S., & Zaki, M. (2006, April). Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security* (Vol. 30, pp. 798-805).

Alonso Casero, Á. (2021). *Named entity recognition and normalization in biomedical literature: a practical case in SARS-CoV-2 literature* (Doctoral dissertation, ETSI_Informatica).

Awasthi, I., Gupta, K., Bhogal, P. S., Anand, S. S., & Soni, P. K. (2021, January). Natural Language Processing (NLP) based Text Summarization-A Survey. In 2021 6th International Conference on Inventive Computation Technologies (ICICT) (pp. 1310-1317). IEEE.

Balakrishnan, V., & Lloyd-Yemoh, E. (2014). Stemming and lemmatization: a comparison of retrieval performances. (pp. 174-179).

Belarbi, M. A., Mahmoudi, S., & Belalem, G. (2017). PCA as dimensionality reduction for large-scale image retrieval systems. *International Journal of Ambient Computing and Intelligence (IJACI)*, *8*(4), 45-58.

Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".

Blendowski, M., Hansen, L., & Heinrich, M. P. (2021). Weakly-supervised learning of multi-modal features for regularised iterative descent in 3D image registration. *Medical image analysis*, *67*, 101822.

Bondielli, A., & Marcelloni, F. (2021). On the use of summarization and transformer architectures for profiling résumés. *Expert Systems with Applications*, *184*, 115521.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, *30*(7), 1145-1159.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, *2*(2), 121-167.

Buscema, M. (1998). Back propagation neural networks. *Substance use & misuse*, *33*(2), 233-270.

Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., Assent I. & Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery*, *30*(4), 891-927.

Cao, L. J., Chua, K. S., Chong, W. K., Lee, H. P., & Gu, Q. M. (2003). A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. Neurocomputing, 55(1-2), 321-336.

Carey, J. (1980, June). Paralanguage in computer mediated communication. In *18th Annual Meeting of the Association for Computational Linguistics* (pp. 67-69).

Carnevale, L., Celesti, A., Fiumara, G., Galletta, A., & Villari, M. (2020). Investigating classification supervised learning approaches for the identification of critical patients' posts in a healthcare social network. *Applied Soft Computing*, *90*, 106155.

Caruana, R., & Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* (pp. 161-168).

Chan, S., & Fyshe, A. (2018, June). Social and emotional correlates of capitalization on twitter. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in social media* (pp. 10-15).

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 1-13.

Chowdhury, G. G. (2003). Natural language processing. *Annual review of information science and technology*, *37*(1), 51-89.

Cunningham, P. (2008). Dimension reduction. In Machine learning techniques for multimedia (pp. 91-112). Springer, Berlin, Heidelberg.

Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia* (pp. 21-49). Springer, Berlin, Heidelberg.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

El-Khair, I. A. (2006). Effects of stop words elimination for Arabic information retrieval: a comparative study. *International Journal of Computing & Information Sciences*, *4*(3), 119-133.

Ferilli, S. (2021). Automatic Multilingual Stopwords Identification from Very Small Corpora. *Electronics*, *10*(17), 2169.

Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, *30*(4), 681-694.

Francis, S., Van Landeghem, J., & Moens, M. F. (2019). Transfer learning for named entity recognition in financial and biomedical documents. Information, 10(8), 248.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *science*, *315*(5814), 972-976.

Garcia, M., & Gamallo, P. (2015, June). Yet another suite of multilingual NLP tools. In *International Symposium on Languages, Applications and Technologies* (pp. 65-75). Springer, Cham.

Ghahramani, Z. (2003, February). Unsupervised learning. In *Summer School on Machine Learning* (pp. 72-112). Springer, Berlin, Heidelberg.

Ghassami, A., Khodadadian, S., & Kiyavash, N. (2018, June). Fairness in supervised learning: An information theoretic approach. In *2018 IEEE International Symposium on Information Theory (ISIT)* (pp. 176-180). IEEE.

Glielmo, A., Husic, B. E., Rodriguez, A., Clementi, C., Noé, F., & Laio, A. (2021). Unsupervised Learning Methods for Molecular Simulation Data. *Chemical Reviews*.

Grefenstette, G. (1999). Tokenization. In *Syntactic Wordclass Tagging* (pp. 117-133). Springer, Dordrecht.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J. & Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. ACM Transactions on Computing for Healthcare (HEALTH), 3(1), 1-23.

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003, November). KNN model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"* (pp. 986-996). Springer, Berlin, Heidelberg.

HaCohen-Kerner, Y., Miller, D., & Yigal, Y. (2020). The influence of preprocessing on text classification using a bag-of-words representation. *PloS one*, *15*(5), e0232525.

Han, J. M., Babuschkin, I., Edwards, H., Neelakantan, A., Xu, T., Polu, S., Ray, A., Shyam, P., Ramesh, A., Radford, A. & Sutskever, I. (2021). Unsupervised Neural Machine Translation with Generative Language Models Only. *arXiv preprint arXiv:2110.05448*.

Hao, L., & Hao, L. (2008, December). Automatic identification of stop words in chinese text classification. In *2008 International conference on computer science and software engineering* (Vol. 1, pp. 718-722). IEEE.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. *Advances in neural information processing systems*, *29*, 3315-3323.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, *28*(1), 100-108.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised learning. In The elements of statistical learning (pp. 485-585). Springer, New York, NY.

Hawkins, D. M. (1980). *Identification of outliers* (Vol. 11). London: Chapman and Hall.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261-266.

Ho, L. N., Tran, A. T., Phung, Q., & Hoai, M. (2021). Toward Realistic Single-View 3D Object Reconstruction With Unsupervised Learning From Multiple Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 12600-12610).

Huang, D., Song, B., Wei, J., Su, J., Coenen, F., & Meng, J. (2021). Weakly supervised learning of RNA modifications from low-resolution epitranscriptome data. *Bioinformatics*, *37*(Supplement_1), i222-i230.

Jamal, A., Handayani, A., Septiandri, A. A., Ripmiatin, E., & Effendi, Y. (2018). Dimensionality reduction using pca and k-means clustering for breast cancer prediction. *Lontar Komput. J. Ilm. Teknol. Inf*, *9*(3), 192.

Jivani, A. G. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, *2*(6), 1930-1938.

Kamani, M. M., Forsati, R., Wang, J. Z., & Mahdavi, M. (2021). Pareto Efficient Fairness in Supervised Learning: From Extraction to Tracing. *arXiv preprint arXiv:2104.01634*.

Kavuluru, R., Rios, A., & Lu, Y. (2015). An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial intelligence in medicine*, *65*(2), 155-166.

Kestemont, M., De Pauw, G., van Nie, R., & Daelemans, W. (2017). Lemmatization for variation-rich languages using deep learning. Digital Scholarship in the Humanities, 32(4), 797-815.

Khodadadian, S., Ghassami, A., & Kiyavash, N. (2021). Impact of Data Processing on Fairness in Supervised Learning. arXiv preprint arXiv:2102.01867.

Khyani, D., Siddhartha, B. S., Niveditha, N. M., & Divya, B. M. (2020). An Interpretation of Lemmatization and Stemming in Natural Language Processing. 22 (10), 350-357.

Kim, M., Yun, J., Cho, Y., Shin, K., Jang, R., Bae, H. J., & Kim, N. (2019). Deep learning in medical imaging. Neurospine, 16(4), 657.

KM, A. K., Kiran, B. R., Shreyas, B. R., & Victor, S. J. (2015). A multimodal approach to detect user's emotion. *Procedia Computer Science*, *70*, 296-303.

Kon, A. (2017). Sobre inovação tecnológica, tecnologia apropriada e mercado de trabalho. (pp. 2).

Korenius, T., Laurikkala, J., Järvelin, K., & Juhola, M. (2004, November). Stemming and lemmatization in the clustering of finnish text documents. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management* (pp. 625-633).

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3-24.

Kumar, G. K., & Rani, D. M. (2021, February). Paragraph summarization based on word frequency using NLP techniques. In AIP Conference Proceedings (Vol. 2317, No. 1, p. 060001). AIP Publishing LLC.

Lagus, J., & Klami, A. (2021). Learning to Lemmatize in the Word Representation Space. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)* (pp. 249-258).

Liddy, E.D. (2001). Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.

Liu, B. (2011). Supervised learning. In *Web data mining* (pp. 63-132). Springer, Berlin, Heidelberg.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

Loh, W. Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *1*(1), 14-23.

Markovic, S., Bryan, J. L., Ishimtsev, V., Turakhanov, A., Rezaee, R., Cheremisin, A., Kantzas, A., Koroteev, D., Mehta, S. A. (2020). Improved Oil Viscosity Characterization by Low-Field NMR Using Feature Engineering and Supervised Learning Algorithms. *Energy & Fuels*, *34*(11), 13799-13813.

Mayhew, S., Nitish, G., & Roth, D. (2020, April). Robust named entity recognition with truecasing pretraining. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 05, pp. 8480-8487).

Melián-González, S., Gutiérrez-Taño, D., & Bulchand-Gidumal, J. (2021). Predicting the intentions to use chatbots for travel and tourism. Current Issues in Tourism, 24(2), 192-210.

Mohit, B. (2014). Named entity recognition. In *Natural language processing of semitic languages* (pp. 221-245). Springer, Berlin, Heidelberg.

Mozafari, M., Farahbakhsh, R., & Crespi, N. (2019, December). A BERT-based transfer learning approach for hate speech detection in online social media. In International Conference on Complex Networks and Their Applications (pp. 928-940). Springer, Cham.

Mullen, L. A., Benoit, K., Keyes, O., Selivanov, D., & Arnold, J. (2018). Fast, consistent tokenization of natural language text. *Journal of Open Source Software*, *3*(23), 655.

Nebhi, K., Bontcheva, K., & Gorrell, G. (2015, May). Restoring capitalization in# tweets. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 1111-1115).

Nettleton, D. F., Orriols-Puig, A., & Fornells, A. (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. Artificial intelligence review, 33(4), 275-306.

Nielsen, F. (2016). Hierarchical clustering. In *Introduction to HPC with MPI for Data Science* (pp. 195-211). Springer, Cham.

Nzeyimana, A. (2020, December). Morphological disambiguation from stemming data. In Proceedings of the 28th International Conference on Computational Linguistics (pp. 4649-4660).

Orasmaa, S., Petmanson, T., Tkachenko, A., Laur, S., & Kaalep, H. J. (2016, May). Estnltk-nlp toolkit for estonian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 2460-2466).

Paice, C. D. (1994). An evaluation method for stemming algorithms. In *SIGIR'94* (pp. 42-50). Springer, London.

Păiş, V., & Tufiş, D. (2021). Capitalization and punctuation restoration: a survey. *Artificial Intelligence Review*, 1-42.

Pak, I., & Teh, P. L. (2018, February). Value of expressions behind the letter capitalization in product reviews. In *Proceedings of the 2018 7th International Conference on Software and Computer Applications* (pp. 147-152).

Park, K., Lee, J., Jang, S., & Jung, D. (2020). An Empirical Study of Tokenization Strategies for Various Korean NLP Tasks. *arXiv preprint arXiv:2010.02534*.

Pham, D. T., Dimov, S. S., & Nguyen, C. D. (2005). Selection of K in K-means clustering. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 219(1), 103-119.

Piernik, M., & Morzy, T. (2021). A study on using data clustering for feature extraction to improve the quality of classification. *Knowledge and Information Systems*, 1-35.

Pinzone, M., Fantini, P., Perini, S., Garavaglia, S., Taisch, M. & Miragliotta, G. (2017). Jobs and Skills in Industry 4.0: An Exploratory Research. IFIP International Conference on Advances in Production Management Systems (APMS), Hamburg, Germany. (pp.282-288)

Plisson, J., Lavrac, N., & Mladenic, D. (2004, May). A rule based approach to word lemmatization. In *Proceedings of IS* (Vol. 3, pp. 83-86).

Powers, D. M. (2014). What the F--measure doesn't measure…. *Technical report, Beijing University of Technology, China & Flinders University, Australia, Tech. Rep.*

Rianto, R., Mutiara, A. B., Wibowo, E. P., & Santosa, P. I. (2021). Improving the Accuracy of Text Classification using Stemming Method, A Case of Non-formal Indonesian Conversation.

Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. 22, pp. 41-46).

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick C. L., Ma J. & Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, *118*(15).

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Ruder, S., Peters, M. E., Swayamdipta, S., & Wolf, T. (2019, June). Transfer learning in natural language processing. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials (pp. 15-18).

Rust, P., Pfeiffer, J., Vulić, I., Ruder, S., & Gurevych, I. (2020). How good is your tokenizer? on the monolingual performance of multilingual language models. *arXiv preprint arXiv:2012.15613*.

Saito, S., Yang, J., Ma, Q., & Black, M. J. (2021). SCANimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2886-2897).

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Sarica, S., & Luo, J. (2021). Stopwords in technical language processing. Plos one, 16(8), e0254937.

Serra, A., & Tagliaferri, R. (2019). Unsupervised Learning: Clustering.

Singh, J., & Gupta, V. (2016). Text stemming: Approaches, applications, and challenges. *ACM Computing Surveys (CSUR)*, *49*(3), 1-46.

Sivakumar, S., Nayak, S. R., Vidyanandini, S., Kumar, J. A., & Palai, G. (2018). An empirical study of supervised learning methods for breast cancer diseases. *Optik*, *175*, 105-114.

Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006, December). Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence* (pp. 1015-1021). Springer, Berlin, Heidelberg.

Song, X., Salcianu, A., Song, Y., Dopson, D., & Zhou, D. (2020). Linear-Time WordPiece Tokenization. *arXiv preprint arXiv:2012.15524*.

Specia, L., & Wilks, Y. (2021). Machine translation. In The Oxford Handbook of Computational Linguistics 2nd edition.

Sun, C., & Yang, Z. (2019, November). Transfer learning in biomedical named entity recognition: An evaluation of BERT in the PharmaCoNER task. In Proceedings of The 5th Workshop on BioNLP Open Shared Tasks (pp. 100-104).

Tong, Y., Sun, Y., Zhou, P., Shen, Y., Jiang, H., Sha, X., & Chang, S. (2021). Locating abnormal heartbeats in ECG segments based on deep weakly supervised learning. *Biomedical Signal Processing and Control*, *68*, 102674.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaise, L., & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

Vinod, V., Agrawal, S., Gaurav, V., & Choudhary, S. (2021). Multilingual Medical Question Answering and Information Retrieval for Rural Health Intelligence Access. arXiv preprint arXiv:2106.01251

Wang, X., Lin, X., & Dang, X. (2020). Supervised learning in spiking neural networks: A review of algorithms and evaluations. *Neural Networks*, *125*, 258-280.

Weber, L., Sänger, M., Münchmeyer, J., Habibi, M., Leser, U., & Akbik, A. (2021). HunFlair: an easy-to-use tool for state-of-the-art biomedical named entity recognition. *Bioinformatics*, *37*(17), 2792-2794.

Webster, J. J., & Kit, C. (1992). Tokenization as the initial phase in NLP. In *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*.

Wilbur, W. J., & Sirotkin, K. (1992). The automatic identification of stop words. *Journal of information science*, *18*(1), 45-55.

Wolf T., Debut L., Sanh V., Chaumond J., Delangue C., Moi A., Cistac P., Rault T., Louf R., Funtowicz M., Davison J., Shleifer S., Platen P. V., Ma C., Jernite Y., Plu J., Xu C., Scao T. L., Gugger S., Drame M., Lhoest Q., & Rush A. (2020, October). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 38-45).

Wu, Q., Wan, J., & Chan, A. B. (2021). Progressive Unsupervised Learning for Visual Object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2993-3002).

Xiao, Y., & Yu, J. (2012). Partitive clustering (K-means family). *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(3), 209-225.

Yang, G., Dineen, S., Lin, Z., & Liu, X. (2021, August). Few-Sample Named Entity Recognition for Security Vulnerability Reports by Fine-Tuning Pre-trained Language Models. In *International Workshop on Deployable Machine Learning for Security Defense* (pp. 55-78). Springer, Cham.

Yang, Q., Zhang, Y., Dai, W., & Pan, S. J. (2020). Transfer learning. Cambridge University Press.

Yedla, M., Pathakota, S. R., & Srinivasa, T. M. (2010). Enhancing K-means clustering algorithm with improved initial center. *International Journal of computer science and information technologies*, *1*(2), 121-125.

Yin, Q., Yang, J., Tyagi, M., Zhou, X., Hou, X., & Cao, B. (2021). Field data analysis and risk assessment of gas kick during industrial deepwater drilling process based on supervised learning algorithm. *Process Safety and Environmental Protection*, *146*, 312-328.

Yin, Z., Zhang, C., Goldberg, D. W., & Prasad, S. (2019, March). An NLP-based question answering framework for spatio-temporal analysis and visualization. In *Proceedings of the 2019 2nd International Conference on Geoinformatics and Data Analysis* (pp. 61-65).

Yuan, F., Meng, Z. H., Zhang, H. X., & Dong, C. R. (2004, August). A new algorithm to get the initial centroids. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)* (Vol. 2, pp. 1191-1193). IEEE.

Zanine, N., Dhawan, V, (2015, January). Text Mining: An introduction to theory and some applications (pp. 38-39). Research Matters: A Cambridge Assessment publication.

Zhang, H., Cheng, Y. C., Kumar, S., Chen, M., & Mathews, R. (2021). Position-Invariant Truecasing with a Word-and-Character Hierarchical Recurrent Neural Network. *arXiv preprint arXiv:2108.11943*.

Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017). Efficient kNN classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems*, *29*(5), 1774-1785.

Zhou, Z. H. (2018). A brief introduction to weakly supervised learning. *National science review*, *5*(1), 44-53.