UNIVERSIDADE FEDERAL DO CEARÁ

INSTITUTO UNIVERSIDADE VIRTUAL

CURSO DE GRADUAÇÃO EM SISTEMAS E MÍDIAS DIGITAIS

ULISSES SILVA DE SOUSA

UMA PROPOSTA DE AVALIAÇÃO DE EMBEDDINGS DE PALAVRAS POR SIMILARIDADE

FORTALEZA

2022

ULISSES SILVA DE SOUSA

UMA PROPOSTA DE AVALIAÇÃO DE EMBEDDINGS DE PALAVRAS POR SIMILARIDADE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas e Mídias Digitais do Instituto Universidade Virtual da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas e Mídias Digitais.

Orientadora: Profª. Drª. Ticiana Linhares Coelho da Silva

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S698p Sousa, Ulisses Silva de.
Uma proposta de avaliação de embeddings de palavras por similaridade / Ulisses Silva de Sousa. – 2022.
45 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto UFC Virtual, Curso de Sistemas e Mídias Digitais, Fortaleza, 2022.
Orientação: Profa. Dra. Ticiana Linhares Coelho da Silva.

1. Word embeddings. 2. Processamento de linguagem natural. 3. Aprendizagem profunda. 4. Similaridade. I. Título.

CDD 302.23

ULISSES SILVA DE SOUSA

UMA PROPOSTA DE AVALIAÇÃO DE EMBEDDINGS DE PALAVRAS POR SIMILARIDADE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas e Mídias Digitais do Instituto Universidade Virtual da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas e Mídias Digitais.

Aprovada em: 08 de Dezembro de 2022

BANCA EXAMINADORA

Profª. Drª. Ticiana Linhares Coelho da Silva (Orientadora)
Universidade Federal do Ceará (UFC)

Profª. Drª. Lívia Almada Cruz
Universidade Federal do Ceará (UFC)

Profª. Bel. Bárbara Stéphanie Neves Oliveira
Universidade Federal do Ceará (UFC)

Aos meus pais, Benúzia Silva e Luiz Ferreira. Aos meus irmãos, Felipe Silva e Rafaella Silva. À minha tia, Antônia Neves. Agradeço pelo cuidado, carinho, suporte e investimento que dedicaram a minha pessoa, significou toda a segurança e esperança para que eu seguisse em frente.

AGRADECIMENTOS

Agradeço primeiramente à minha família, em especial a minha mãe Benúzia Silva de Sousa, minha tia Antônia Neves da Silva, meus irmãos Rafaella Silva de Sousa e Felipe Silva de Sousa, meu pai Luiz Ferreira de Sousa Júnior, sem os quais não seria possível todo esse trajeto de graduação. Agradeço a todas as partes da minha família, minha avó, tias e tios, primos e primas, sobrinho.

Um agradecimento especial aos que convivem comigo, minha mãe, tia e irmão, por proporcionarem um ambiente adequado de estudos, aporio financeiro, incentivos constantes e torcida incondicional. Por suportarem meus piores momentos e lidarem com minha ausência em momentos importantes.

Ao meu cunhado Prof. Dr. Pablo Mayckon Silva Farias por ter sido um dos primeiros a me incentivar a voltar aos estudos, pelas conversas sobre a UFC, pelo entusiasmo das discussões políticas, por demonstrar interesse em minha carreira acadêmica e por torcer junto com minha família pelo meu sucesso.

À Profª. Drª. Ticiana Linhares Coelho da Silva por me orientar com tanta paciência e sabedoria, por ter sido uma das maiores fontes de inspiração em minha carreira acadêmica, pelo incentivo aos estudos na área de sistemas e por ter me iniciado nos estudos de *Machine Learning* e *Deep Learning*. Agradeço também por ter sido sincera e resoluta nos momentos que mais precisei de firmeza e, principalmente, por não ter desistido de mim quando estava desestimulado com a formação acadêmica, por ter ajudado a superar a minha falta de confiança no meu próprio potencial. Obrigado por ser essa pessoa inteligente, gentil e forte que tanto admiro.

À Profª Bárbara Stéphanie Neves Oliveira que foi uma grande mentora, tanto em assuntos acadêmicos como profissionais, pelos momentos de desabafo, pela amizade e também pelo seu TCC que foi fonte de inspiração para os meus estudos.

Aos integrantes do laboratório *Insight* da UFC, que, embora o contato tenha sido breve, foram cruciais para a minha formação na área de Processamento de Linguagem Natural, principalmente nas produções acadêmicas de seus integrantes que serviram de fonte de pesquisa e inspiração.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

Agradeço aos meus prezados colegas de trabalho pelo apoio moral e descontrações que ajudaram a encarar essa etapa final da minha formação.

Ao amigo e chefe João Lucas de Oliveira Timbó pelo acolhimento profissional, por ter arriscado investir no meu potencial, por me ensinar a ser mais objetivo e pelo incentivo à conclusão da minha graduação.

A todos os meus amigos íntimos pela paciência de escutar meus monólogos sobre a graduação e meu trabalho, pelos momentos de lazer e descontração que foram importantíssimos para renovar meu ânimo e perseverança.

E a todos os professores do Sistemas e Mídias Digitais por me proporcionarem o conhecimento não apenas racional, mas na manifestação de caráter e afetividade como educadores, por terem me desafiado e me ajudado a me encontrar no curso.

“Ensinar não é transferir conhecimento, mas criar as possibilidades para a sua própria produção ou a sua construção.”

(Paulo Freire)

RESUMO

A representação de texto humano em informações numéricas é uma tarefa difícil que tem sido investigada e utilizada no campo da Inteligência Artificial. Nesse contexto, o uso de representações vetoriais através de *Word Embeddings* e *Sentence Embeddings* para tarefas de Processamento de Linguagem Natural (*Natural Language Processing* em inglês) tornou-se muito comum no estado da arte, é também um conhecimento muito estudado na área de ciências de dados. Usá-los como camadas ocultas em modelos de Aprendizagem Profunda reduz muito os custos e melhora o desempenho da maioria das tarefas. Contudo, como existem muitos desses algoritmos e *embeddings* pré-treinadas, não é uma escolha óbvia para um cientista de dados qual modelo se encaixa melhor em seu contexto. Portanto, muitas estratégias buscam avaliar o desempenho de tais modelos, intrínseca e extrinsecamente, para que alguma intuição ou conclusão sobre eles possa ser utilizada para auxiliar no processo de criação de um modelo estatístico ou de Aprendizagem Profunda em Processamento de Linguagem Natural. Assim, buscando desenvolver uma forma de avaliar intrinsecamente alguns dos *Word Embeddings* clássicos, este trabalho propõe uma avaliação de *Word Embeddings* pré-treinados por meio da similaridade de palavras, investigando a semelhança semântica e associação entre palavras aprendidas pelos *embeddings* e baseando-os em conjuntos de dados pré-anotados de palavras sinônimas.

Palavras-chave: Word Embeddings. Processamento de Linguagem Natural. Aprendizagem Profunda. Similaridade.

ABSTRACT

The representation of human text in numerical information is a difficult task that has been investigated and used on the field of Artificial Intelligence. In this context, the use of vector representations through Word Embeddings and Sentence Embeddings for Natural Language Processing tasks has become very common in the state of art, it is also a well-studied knowledge in the area of data sciences. Using them as hidden layers in Deep Learning models greatly reduces costs and improves performance for most tasks. However, as there are many such algorithms and pre-trained embeddings, it is not an obvious choice for a data scientist which model best fits their context. Therefore, many strategies seek to evaluate the performance of such models, intrinsically and extrinsically, so that some intuition or conclusion about them can be used to assist in the process of creating a statistical or Deep Learning model in Natural Language Processing. Thus, seeking to develop a way to intrinsically evaluate some of the classic Word Embeddings, this work proposes an evaluation of pre-trained Word Embeddings through word similarity, investigating the semantic similarity and association between learned words by embeddings and basing them on pre-annotated datasets of synonym words.

Keywords: Word Embeddings. Natural Language Processing. Deep Learning. Similarity.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

GloVe	*Global Vectors for Word Representation*
NLP	*Natural Language Processing*
tSNE	*T-distributed Stochastic Neighbourhood Embedding*
IA	Inteligência Artificial
PLN	Processamento de Linguagem Natural

SUMÁRIO

1 INTRODUÇÃO

O convívio com as novas tecnologias tem se tornado cada vez mais natural para a humanidade ao longo das últimas décadas, e uma das principais formas de tecnologia que viabiliza as praticidades cotidianas é o Processamento de Linguagem Natural (PLN), ou *Natural Language Processing* (*NLP*) em inglês. Aplicações que usam Assistentes de Voz e Checagem Gramatical, por exemplo, são possíveis através dos modelos de PLN que realizam tarefas como Recuperação de Informação, Classificação de Texto e Sumarização de Texto (OTTER *et al.*, 2020).

O PLN é uma das áreas dentro do grande campo de Inteligência Artificial (IA) que liga os processos da linguagem do ser humano com o processamento das máquinas, ou seja, com base na linguagem natural e escrita, ou mesmo a falada, o computador pode analisar, compreender e responder tarefas que lhe são dadas. À primeira vista pode parecer simples, mas quando se lida com uma máquina que só é capaz de entender zeros (0's) e uns (1's), traduzir para o computador um sistema complexo como a linguagem humana, que é cheia de subjetividade, é uma tarefa nada óbvia e difícil. No caso das tarefas de PLN, que se dedicam a geração e compreensão automática de textos escritos por humanos, geralmente são implementadas através de algoritmos de Aprendizagem de Máquina (*Machine Learning* em inglês) ou de Aprendizagem Profunda (*Deep Learning* em inglês), onde a máquina simula a maneira como os humanos aprendem para automática e gradualmente melhorar a sua performance na tarefa que se propõe (CHOLLET, 2018).

A representação padrão de palavras, textos e caracteres por meio de bits não é suficiente para capturar aspectos semânticos, sintáticos, associações e estabelecer conexão entre as palavras e sentenças de um texto. Então, um dos problemas enfrentados pelos cientistas, engenheiros ou analistas de dados, é justamente como representar essa linguagem humana para o computador de forma a se estabelecer relação entre elas. Destarte, uma das estratégias mais famosas para representação textual são através do uso de *Word Embeddings* e *Sentence Embeddings*, que são formas de representação matemática dentro de um espaço vetorial para palavras e sentenças completas de um texto, respectivamente. Sua maior vantagem é permitir relacionar as palavras e sentenças dentro dos diferentes textos do seu conjunto de dados (CHOLLET, 2018). Pode-se encontrar várias dessas representações na literatura, e seu uso tornou-se tão popular nas tarefas de PLN que existem hoje diferentes estratégias de construção de *Word Embeddings* (TORREGROSSA *et al.*, 2021). Além disso, um uso comum no estado da arte são as *embeddings*

pré-treinadas, que se tratam de modelos que já foram treinados dentro de um contexto mais genérico, servindo para melhorar os resultados de um novo modelo de Aprendizagem Profunda que pretende-se treinar, sendo inseridas como uma camada à mais durante o treinamento e diminuindo os custos envolvendo a computação do modelo proposto (QI *et al.*, 2018).

No entanto, embora a estratégia de usar *Sentence Embeddings* e *Word Embeddings* seja bastante popular no estado da arte, os cientistas de dados ainda se deparam com a questão de qual seria, dentre os *embeddings* existentes, a melhor forma de representação para o seu problema específico (JÚNIOR *et al.*, 2021). Não é tão óbvio qual deles usar, pois os algoritmos podem variar a performance de acordo com a tarefa e o contexto dos dados, e faz-se necessário também comparar e visualizar como esses *embeddings* estão representando as palavras para que o cientista em questão possa ter uma intuição mais objetiva em relação ao seu problema.

Os métodos atuais de avaliação podem ser classificados em duas categorias, intrínseca e extrínseca: na avaliação intrínseca, a análise é feita no próprio *embedding* verificando similaridades, analogias e utilizando plotagens para checar a captura semântica ou se a informação foi aprendida do modo esperado. No caso da avaliação extrínseca, consiste em utilizar o *embedding* na execução de alguma tarefa para checar se a tarefa em questão obteve uma melhoria na performance por conta do uso daquele *embedding* (SCHNABEL *et al.*, 2015).

Muitos trabalhos propõem de maneiras diferentes avaliações extrínseca ou intrínseca de modelos e *embeddings*. Por exemplo, no trabalho de Toshevska *et al.* (2020), o parâmetro de avaliação é através da similaridade cosseno entre os vetores de palavras, utilizando a média desses valores como fator de comparação. E, como recurso de prova na hora de fazer a avaliação, são utilizados *benchmarks*, que na computação podem ser entendidos como métodos de comparação de sistemas, subsistemas ou arquiteturas (GRAY, 1993). No caso da avaliação de *Word Embeddings*, os *benchmarks* em questão são conjuntos de dados de palavras pré-anotadas com alguma informação semântica ou sintática dada por humanos. A Tabela 1 traz um excerto do *benchmark Card660*, onde os dados são anotados em pares de palavras com uma pontuação de similaridade entre cada par, variando de zero (pouco similar) à quatro (bastante similar).

Sendo assim, tendo em vista a problemática envolvendo a escolha de *Word Embeddings* pré-treinadas, o objetivo geral deste trabalho é propor uma avaliação das palavras aprendidas por alguns *Word Embeddings* comparando com *benchmarks* pré-anotados de similaridade entre palavras. No entanto, devido a escassez de *benchmarks* no idioma português, a proposta deste trabalho se limitará aos *benchmarks* e *Word Embeddings* no idioma inglês, apesar

Tabela 1 – Excerto do *benchmark Card660*

Palavra 1	Palavra 2	Pontuação
Pokemon	Pocket Monsters	3.81
prejudice	chauvinist	2.25
formic acid	arachnology	1.19
full-HD	1080p	4.00
convocation	gathering	3.56

Fonte: elaborado pelo autor (2022).

de existirem *embeddings* pré-treinadas em português não seria possível estabelecer a prova de comparação sem os *benchmarks*. Tendo como objetivos específicos, a investigação intrínseca sobre a similaridade de palavras de *embeddings* pré-treinadas, extrair informações sobre os algoritmos de treinamento e fazer um ranqueamento do desempenho de cada *Word Embedding* em relação aos *benchmarks*.

Os próximos capítulos estão organizados da seguinte maneira: o Capítulo 2 apresenta a fundamentação teórica e os conceitos que embasam as abordagens propostas neste trabalho; o Capítulo 3 trata dos trabalhos relacionados que foram fonte de inspiração para a problemática aqui proposta; o Capítulo 4 traz a metodologia utilizada, com descrição dos passos necessários que foram usados para atingir o objetivo do trabalho; no Capítulo 5 encontram-se os resultados obtidos pela avaliação proposta; e o Capítulo 6 traz as conclusões deste trabalho e discute trabalhos futuros.

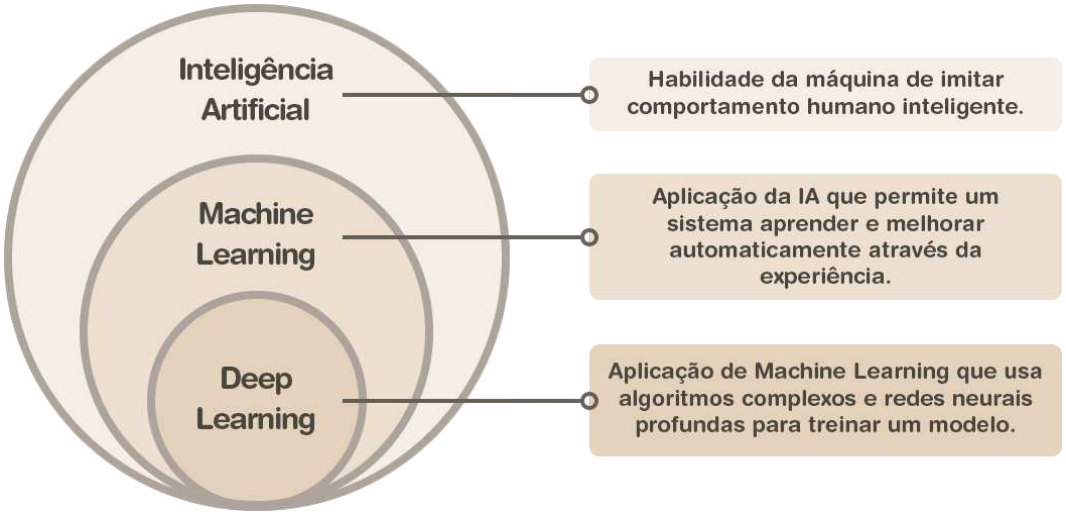
2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está dividido em três seções. A primeira seção aborda os conceitos essenciais para o desenvolvimento teórico deste trabalho, definindo as áreas de IA, Aprendizagem de Máquina e Aprendizagem Profunda. Em seguida, a segunda seção apresenta a área de PLN e também descreve e detalha as técnicas de *Word Embeddings* e *Sentence Embeddings* mais usadas na literatura. A terceira seção fala sobre a similaridade de palavras, que é o ponto essencial deste trabalho na avaliação das *embeddings* pré-treinadas.

2.1 Inteligência Artificial, Aprendizagem de Máquina e Aprendizagem Profunda

Primeiramente, é necessário entender algumas terminologias comuns quando o assunto é IA, Aprendizagem de Máquina e Aprendizagem Profunda. E para compreender melhor como os três (3) assuntos se relacionam, é possível agrupá-los visualmente em uma área dentro da outra como mostra a Figura 1.

Figura 1 – Relação entre as áreas de IA



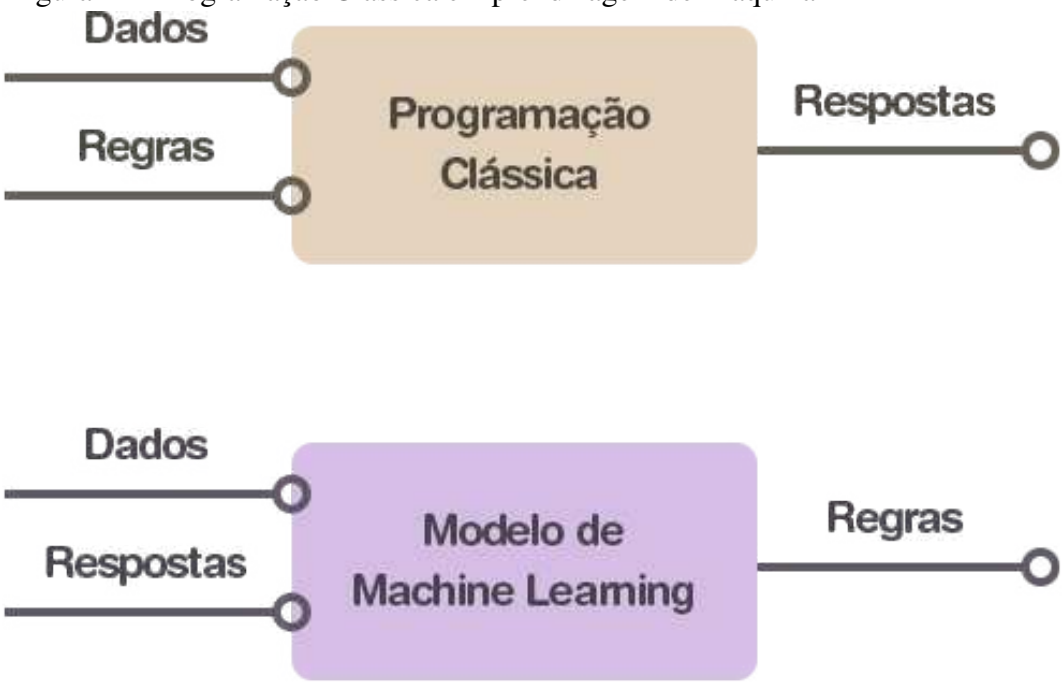
Fonte: elaborado pelo autor (2022).

Chollet (2018) traz conceitos bem definidos e concisos sobre cada uma dessas áreas. Começando pelo campo maior de IA, ela pode ser definida como um esforço de automatizar tarefas intelectuais que normalmente são feitas por seres humanos, podendo ser aplicada em vários níveis, desde problemas mais simples como mais complexos. É, então, um campo mais geral onde estão inseridos os de Aprendizagem de Máquina e Aprendizagem Profunda. Uma rotina que simula um ser humano jogando uma partida de damas, por exemplo, é considerada uma aplicação de IA por tentar simular uma tarefa inerentemente humana. Esse tipo de abor-

dagem é chamada de IA simbólica, pois é manualmente programada para resolver uma tarefa específica. No entanto, problemas mais complexos, que exigem um maior grau de subjetividade e interpretação, como Reconhecimento de Imagens ou Tradução Automática começaram a ser possíveis de solucionar graças a Aprendizagem de Máquina ((CHOLLET, 2018)).

Aprendizagem de Máquina significa fazer a máquina aprender por conta própria simulando a maneira como seres humanos aprendem, tal qual um ser humano aprende por si só a andar, por exemplo. Segundo (CHOLLET, 2018), essa ideia de aprendizado veio justamente do questionamento se o computador seria capaz de aprender a resolver uma determinada tarefa por conta própria olhando e analisando dados sobre aquela tarefa. E foi assim que nesse questionamento abriu-se caminho para um novo paradigma de programação. Se considerarmos a programação clássica, onde o programador faz as regras e coloca dados de entrada para a máquina e espera assim obter determinadas respostas, com Aprendizagem de Máquina o processo é quase que inverso, o computador recebe dados de entrada junto com as respostas referentes a esses dados, e é esperado que a partir disso ele aprenda regras de como resolver essa tarefa em questão (vide Figura 2).

Figura 2 – Programação Clássica e Aprendizagem de Máquina



Fonte: elaborado pelo autor (2022).

Resumindo, Aprendizagem de Máquina é um sistema que é treinado, em vez de explicitamente programado. Por exemplo, considere um conjunto de dados que possua imagens diversas de cachorros e gatos, onde cada imagem está rotulada indicando se o animal presente

na imagem é um cachorro ou gato. Esse conjunto de dados poderia ser usado para treinar um modelo que irá criar suas próprias regras de forma que, dada uma nova imagem de entrada de um cachorro ou gato, com base nos padrões que ele aprendeu, esse modelo irá prever qual o rótulo daquela imagem, se é a palavra “cachorro"ou "gato”. Esse é um exemplo clássico de uma tarefa de Classificação de Imagem.

Aprendizagem Profunda, por sua vez, é uma subárea específica dentro de Aprendizagem de Máquina que usa um procedimento baseado em camadas sucessivas de representação dos dados (CHOLLET, 2018). Os modelos mais modernos utilizam dezenas ou mesmo centenas dessas camadas e também são conhecidos como Redes Neurais, por se inspirar no funcionamento do cérebro humano, com centenas de nós se conectando. A maior diferença está na quantidade de representações que o modelo pode aprender, logo, aumenta consideravelmente a complexidade de informações que podem ser extraídas dos dados passados. Voltando ao exemplo da Classificação de Imagem, no caso de uma estratégia com Aprendizagem Profunda, o modelo poderia ser treinado para prever mais características ou recursos (*features* em inglês) além de classificar o rótulo do animal, como cor dos olhos, número de patas ou expessura do rabo.

2.2 Processamento de Linguagem Natural

Segundo Otter *et al.* (2020), PLN é um campo orientado a dados que usa da estatística e computação probabilística para construir modelos e processos computacionais que resolvem problemas práticos dentro do entendimento da linguagem humana. É com esses modelos e soluções que são criados os diversos softwares que lidam com a linguagem humana, como Tradutores, Auto Corretores de Texto, e Reconhecimento de Fala.

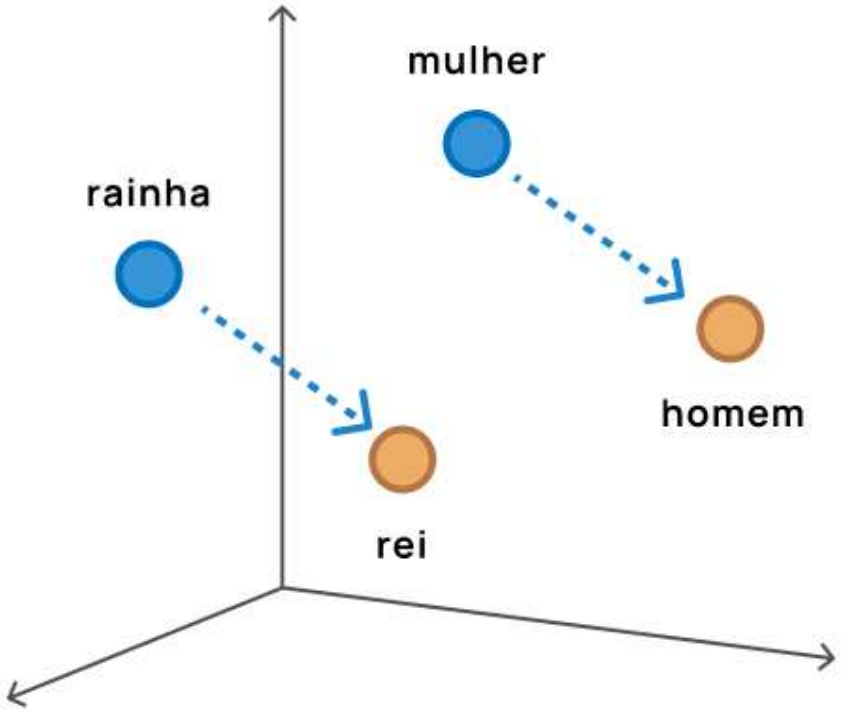
Atualmente, devido ao grande poder computacional das novas tecnologias, o mais comum no estado da arte é ver as estratégias de Aprendizagem Profunda sendo usadas para resolver os problemas de PLN. As primeiras estratégias que usavam algoritmos de Aprendizagem de Máquina, como *Naïve Bayes*, *k-nearest neighbors*, e Árvores de Decisão, durante os últimos anos, foram quase que completamente substituídas pelas Redes Neurais da Aprendizagem Profunda com suas múltiplas camadas de densidade (OTTER *et al.*, 2020).

Como parte da estratégia de resolução de problemas de PLN para representação de palavras ou textos, é muito popular o uso de *Word Embeddings* e *Sentence Embeddings*, assuntos que já possuem um grande material produzido, bem como uma série de estudos relacionados.

2.2.1 Word Embeddings

No caso das *Word Embeddings* clássicas ou estáticas, as quais estão no escopo deste trabalho, são um tipo de representação de palavras através de um espaço vetorial que transforma o texto, ou seja, a linguagem humana é transformada em vetores numéricos de posição fixa (JÚNIOR *et al.*, 2021). Como discutido anteriormente, é preciso que a máquina entenda os dados que lhe são passados de forma que a captura da informação possa estabelecer alguma conexão entre elas. Sendo assim, dependendo do modelo de *embedding* usado, pode-se transformar as sentenças, as palavras ou o documento inteiro em vetores para que possam ser usados nos modelos de Aprendizagem Profunda ou Aprendizagem de Máquina e estabelecer relações de contexto entre os dados, permitindo, por exemplo, identificar similaridades semânticas e sintáticas entre as palavras. A Figura 3 mostra um exemplo de espaço vetorial de três (3) dimensões contendo quatro (4) palavras, a imagem é uma simulação de como seria possível estabelecer relações entre as palavras ao serem representadas por vetores, no caso, os pontos coloridos representam os vetores e pode-se notar que o par de palavras "mulher"e "homem"possui uma relação semelhante ao par "rainha"e "rei".

Figura 3 – Simulação de Espaço Vetorial de 3 dimensões



Fonte: elaborado pelo autor (2022).

Contudo, as *Word Embeddings* normalmente são multidimensionais indo de cinquenta (50) até quinhentas (500) dimensões, o que diretamente aumenta o custo do processamento e também o capacidade de capturar significado das palavras.

Word Embeddings possuem diferentes tipos, podendo ser agrupados de acordo com o algoritmo de implementação usado (OLIVEIRA, 2020). Este trabalho, no entanto, se restringe a três estratégias significativas no estado da arte, sendo elas: *Word2Vec*, *FastText* e *Global Vectors for Word Representation* (*GloVe*). As três (3) *embeddings* citadas são do tipo estáticas, ou seja, com vetores de posição fixa.

Um dos primeiros métodos de *Word Embedding* usado foi o *GloVe*, que tenta identificar analogias sintáticas e semânticas das palavras através do uso de matrizes (PENNINGTON *et al.*, 2014), foi também fonte de inspiração para as estratégias e técnicas sucessoras.

O *Word2Vec* é uma técnica clássica de *Word Embedding*, obteve popularidade quando foi lançado e pode-se ver com frequência seu uso em problemas de PLN. O seu intuito é permitir capturar informações semânticas das palavras a partir dos contextos (HARTMANN *et al.*, 2017a).

Já o *FastText*, desenvolvido por Bojanowski *et al.* (2017), as palavras são representadas usando uma soma de vetores. Como o nome sugere, é um método rápido e eficiente que permite que o modelo aprenda detalhes morfológicos das palavras, obtendo resultados competitivos com a técnica anteriormente citada.

Para o escopo deste trabalho, são avaliadas somente *Word Embeddings* pré-treinadas, que são modelos treinados em um grande conjunto de dados e salvos como vetores multidimensionais, formando assim um grande espaço vetorial que pode ser usado para resolver outras tarefas de PLN, é uma forma de transferir um aprendizado para outra tarefa.

2.2.2 Sentence Embeddings

As *Sentence Embeddings*, por sua vez, possuem uma função similar aos *Word Embeddings*, uma vez que também são representações em um espaço vetorial. No entanto, elas buscam representar exclusivamente as sentenças dos textos, de maneira que palavras similares ou relacionadas semanticamente possam ser consideradas para uma representação contextual (JÚNIOR *et al.*, 2021).

Essa estratégia permite estabelecer conexões entre contexto e palavras, ou palavras e sentenças, de forma que a interpretação do modelo torna-se bem menos literal e mais parecida com a interpretação humana, uma vez que, por exemplo, um ser humano quando exerce a

atividade de leitura de um texto, não se detém apenas na interpretação de cada palavra por palavra, é necessário contextualizar o todo no ato da leitura.

Assim como os *Word Embeddings*, os *Sentence Embeddings* também possuem diferentes tipos e formas de implementação. O *Doc2Vec*, desenvolvido inicialmente por Le e Mikolov (2014), é usado para converter um documento de texto em um vetor, embora não permita fazer uma relação semântica, ele pode ser usado para problemas como identificação de plágio entre documentos ou em ferramentas de busca por artigos similares. Outro método é o *SentenceBERT*, atualmente considerado como o novo estado da arte, já que consegue uma alta performance ao obter similaridade entre sentenças, permitindo, por exemplo, a recuperação de informação através de uma busca semântica (REIMERS; GUREVYCH, 2019).

Outra técnica que possui uma ótima performance é o *Universal Sentence Encoder*, seu recurso principal é poder ser usado em conjunto de dados limitados (CER *et al.*, 2018). Já o modelo proposto por Chidambaram *et al.* (2018) usa essa arquitetura do *Universal Sentence Encoder* para criar um novo modelo que busca englobar sentenças para serem usadas para aprendizado multitarefa, significa que um mesmo *Sentence Embedding* gerado pelo modelo pode ser usado em tarefas distintas como, Análise de Sentimento, Classificação de Texto ou Similaridade de Sentenças.

Não é do conhecimento do autor nenhum *benchmark* de sentenças similares que possa ser utilizado neste trabalho, dessa forma, os *Sentence Embeddings* não serão abordados na avaliação que este trabalho pretende realizar.

2.3 Similaridade de palavras

A avaliação pretendida neste trabalho baseia-se na similaridade entre palavras. Para tanto, faz-se necessário o uso de *benchmarks* pré-anotados de similaridade de palavras. No caso deste trabalho, os *benchmarks* usados são conjuntos de dados que pontuam a semelhança entre pares de palavras, estabelecendo valores entre elas que indicam o quão semelhante são, mas não necessariamente sinônimas. O intuito é que esses pares sejam usados como fator de prova para executar uma operação que compare se a similaridade entre as palavras aprendidas pelas *Word Embeddings* condizem com a similaridade anotada nos *benchmarks*.

Portanto, em vista desse modo de avaliação pretendido, é importante entender o conceito do que é similaridade entre palavras e diferenciar de associação entre palavras. A similaridade a que este trabalho refere-se pode ser entendida como a semelhança semântica

entre duas palavras, ou seja, o quão próxima uma palavra está de outra em seu significado. Inicialmente parece um conceito simples, mas computacionalmente não é tão trivial, pois o que define essa relação é algo subjetivo, trata-se de uma capacidade que os humanos possuem de intuitivamente capturar essas semelhanças. A associação, entretanto, pode ser entendida como o contexto existente entre palavras.

Para deixar mais claro essa distinção, assim como o exemplo de Hill *et al.* (2015), considere os seguintes pares de palavras: [lápis, caneta] e [lápis, borracha]. O lápis é similar (semanticamente) à caneta e associado (não similar) à borracha. O lápis e a caneta podem ser entendidos como semelhantes devido às características que possuem em comum, a função de escrever, a forma física alongada, a categoria e contexto em que podem ser usados, escritório, material escolar, etc. O lápis e a borracha são associados por estarem frequentemente juntos no seu uso, pois possuem uma clara relação funcional, ou seja, o contexto em que aparecem aproxima uma palavra da outra. Importante notar que essa associação, trata-se da proximidade entre essas palavras nos espaços vetoriais das *Word Embeddings* mesmo que seu significado semântico seja diferente. No exemplo dado acima, especula-se que, no espaço vetorial de um modelo de *Word Embedding*, os vetores das três palavras citadas poderiam estar próximos mesmo que elas não possuam uma relação de similaridade entre si, pois as redes neurais que aprendem as *embedding* buscam prever o contexto em que as palavras aparecem, calculando a probabilidade de que certas palavras apareçam juntas dentro de um determinado contexto (BENGIO *et al.*, 2000; MIKOLOV *et al.*, 2013).

3 TRABALHOS RELACIONADOS

Os trabalhos apresentados a seguir apontam o contexto no qual surgiu a proposta deste trabalho. Este capítulo está dividido em duas seções, na primeira seção são abordados trabalhos que avaliam e comparam a qualidade das representações por meio de *embeddings*, onde são discutidas diferentes técnicas de avaliação e comparação de performance dentro de seus respectivos problemas; a segunda seção apresenta trabalhos que propõem conjuntos de dados anotados de similaridade de palavras para serem usados em tarefas de avaliação de performance de modelos de PLN.

3.1 Avaliação e comparação da qualidade das representações via *embeddings*

Um modelo de *Word Embeddings* pode ser avaliado de forma intrínseca ou extrínseca. A avaliação intrínseca ocorre quando o próprio modelo é avaliado, usando alguma regra, estratégia ou parâmetro para discernir o quão bom foram os resultados do seu treinamento. Já no caso da avaliação extrínseca, o modelo é usado como entrada em alguma tarefa de PLN para depois avaliar quão bom foi o resultado da tarefa com a utilização do modelo em questão.

Considerando as formas de avaliação, os trabalhos de Cer *et al.* (2018), Toshevska *et al.* (2020), Júnior *et al.* (2021), Firmiano e Silva (2021) possuem propostas e objetivos diferentes, contudo, em todos são discutidas e comparadas as diferentes técnicas de *embeddings* como parte da resolução de seus problemas.

Cer *et al.* (2018) traz como contribuição uma nova proposta de arquitetura que possui alta performance mesmo com um conjunto de dados escassos. Para tanto, os autores fazem uma avaliação comparativa usando as outras técnicas de *embeddings* para validar a melhoria e uso de recursos da arquitetura proposta. No entanto, falta uma comparação visual dentro do próprio espaço vetorial dos *embeddings*, de forma que seja possível analisar as relações entre as palavras de uma determinada *Word Embedding* ou perceber possíveis enviesamentos no aprendidos pelo modelo.

Tendo em vista a avaliação intrínseca, Toshevska *et al.* (2020) em um de seus métodos comparativos faz a análise de vários modelos de *Word Embeddings* usando a similaridade cosseno entre vetores como fator de comparação com *benchmarks* conhecidos no estado da arte, como o *SimLex999*, *WordSim353* e *SimVerb3500*. Para cada par de palavras dos conjuntos de dados dos *benchmarks*, é calculada a similaridade com as representações daquelas palavras nos modelos

pré-treinados e feita uma média para saber qual modelo chega mais perto da média anotada por humanos nos *benchmarks*.

Júnior *et al.* (2021) busca avaliar de forma extrínseca os diferentes *Sentence Embeddings* e *Word Embeddings* para o problema de descobrir intenções em diálogos sobre COVID-19, além de propor um modelo de entendimento de linguagem natural para tais diálogos.

Firmiano e Silva (2021) investigam maneiras de reconhecer documentos duplicados a nível de sentenças. Um dos objetivos desse trabalho é a avaliação intrínseca de modelos de *embeddings* na tarefa de captura de narrativas duplicadas. No caso, foram usados boletins de ocorrência para formar o *corpus* (coleção de textos autênticos organizados em um conjunto de dados com um contexto determinado pelos autores). São avaliados tanto os modelos pré-treinados, diferentes *Word Embeddings* e *Sentence Embeddings*, como também *embeddings* treinadas pelos próprios autores a partir do vocabulário de boletins de ocorrência.

3.2 *Benchmarks* para avaliação de modelos de *embeddings*

Os dois trabalhos a seguir apresentam novos recursos de conjunto de dados anotados que envolvem a similaridade entre palavras. Em ambos os estudos, o intuito é que os conjuntos de dados possam ser usados em tarefas de avaliação de performance de modelos no idioma inglês.

No trabalho de Hill *et al.* (2015) é apresentado o *SimLex999*, um *benchmark* no idioma inglês que os autores consideram como um sucessor melhorado de outros conjuntos previamente publicados, como o *WordSim353* de Finkelstein *et al.* (2001). O ponto forte do *SimLex999* é que ele explicitamente quantifica a similaridade entre palavras em vez de capturar a associação ou parentesco entre elas. Outra vantagem é a diversidade dos pares de palavras, contendo adjetivos concretos e abstratos, pronomes e verbos.

A abordagem de Inohara e Utsumi (2022) trata-se de um problema mais específico: a falta de dados anotados no idioma japonês, o que dificulta executar tarefas de PLN nesse idioma. Em seu estudo, eles apresentam o *JWSAN* (*Japanese Word Similarity and Association Norm*), um conjunto de dados com pares de palavras contendo valores de similaridade e associação no idioma japonês[1]. A maior vantagem desse conjunto de dados é poder ser usado como *benchmark* para avaliar e melhorar modelos semânticos em japonês.

[1] Disponível em: <http://www.utm.inf.uec.ac.jp/JWSAN/en/>

4 METODOLOGIA

Este trabalho propõe a avaliação de três *Word Embeddings* pré-treinados, sendo eles populares no estado da arte para tarefas de PLN. Adotou-se também o uso de *benchmarks* de similaridade entre palavras que servem de prova para poder avaliar como as *embeddings* aprenderam as semelhanças entre palavras.

As avaliações foram realizadas através do uso do produto *Jupyter Notebook*, um ambiente computacional *web* apropriado para computações nas áreas de Ciências de Dados e Aprendizagem de Máquina.

Devido à falta de benchmarks em português com pares anotados de similaridade entre palavras, decidiu-se usar os conjuntos de dados em inglês mais usados no estado da arte mencionados em Inohara e Utsumi (2022). Portanto, para poder fazer a comparação entre as palavras do conjunto anotado e as palavras do espaço vetorial das *Word Embeddings*, também fez-se necessário usar modelos pré-treinados no idioma inglês de contexto genérico, ou seja, as *embeddings* não foram treinadas em domínios específicos.

A próxima parte deste capítulo apresenta os procedimentos metodológicos para atingir os objetivos da proposta deste trabalho. As etapas do procedimento estão descritas de forma detalhada nas seguintes seções: Coleta de dados, Pré-processamento dos dados, Comparação dos modelos e *benchmarks* e Visualização dos espaços vetoriais.

4.1 Coleta de dados

O primeiro procedimento consistiu em coletar os dados para a avaliação. Para tanto, foi necessário baixar os arquivos definidos como escopo deste trabalho, sendo eles: três (3) *Word Embeddings* pré-treinadas e mais três (3) *benchmarks* de similaridade de palavras. Tanto as *Word Embeddings* como os *benchmarks* se encontram no idioma inglês.

As três *Word Embeddings* pré-treinadas escolhidas para a avaliação intrínseca foram as seguintes: *Word2Vec*, com espaço vetorial de três milhões de palavras (HARTMANN *et al.*, 2017a)[1]; *FastText*, dois milhões de palavras no espaço vetorial (BOJANOWSKI *et al.*, 2017)[2]; *GloVe*, com quatrocentos mil palavras em seu espaço vetorial (PENNINGTON *et al.*, 2014)[3].

As *embeddings* escolhidas são estratégias clássicas de *Word Embeddings* de posição

[1] Disponível em: <https://code.google.com/archive/p/word2vec/>
[2] Disponível em: <https://fasttext.cc/docs/en/english-vectors.html>
[3] Disponível em: <https://nlp.stanford.edu/projects/glove/>

fixa. Cada uma faz uso de vetores de trezentas (300) dimensões para representação das palavras e foram treinadas em domínio de contexto genérico, ou seja, o intuito é de que possam ser usadas em qualquer tarefa de PLN independentemente de seu contexto.

Para completar a coleta, foram usados também os seguintes *benchmarks*: *Card660*, conjunto de dados com seiscentos e sessenta (660) pares de palavras infrequentes, fazendo um total de mil trezentas e vinte (1320) palavras dentro do conjunto (PILEHVAR *et al.*, 2018), o diferencial deste *benchmark* está no uso de palavras consideradas raras em relação aos vocabulários mais comuns[4]; *SimLex999*, o maior dos três, possuindo novecentos e noventa e nove (999) pares de palavras, com um vocabulário total de mil novecentas e noventa e oito (1998) palavras de contexto genérico (HILL *et al.*, 2015)[5]; *VerbPair130*, com apenas cento e trinta (130) pares de palavras, sendo todas elas verbos, o conjunto possui um total de duzentas e sessenta (260) palavras em seu vocabulário (YANG; POWERS, 2015)[6].

Os *benchmarks* escolhidos possuem tamanhos variados e contextos diferentes, com essa escolha supõe-se que as *embeddings* tenham resultados similares, pois as *embeddings* escolhidas, por terem sido treinadas em domínio genérico, independem do contexto. Sobre a similaridade entre palavras dos *benchmarks*, é estabelecido um valor de pontuação para medir a similaridade léxica de duas palavras, no caso do conjunto de dados *Card660* e *VerbPair130*, o valor varia de zero (0) a quatro (4), onde zero significa que o par de palavras não são similares e o quatro significa que são muito similares (praticamente sinônimas). O *SimLex999* utiliza a mesma estratégia, porém os valores variam de zero a dez (10).

4.2 Pré-processamento dos dados

Neste procedimento, para que a comparação fizesse sentido, foi usada uma lógica de pré-processamento para tratar os dados que fariam parte do conjunto a ser avaliado.

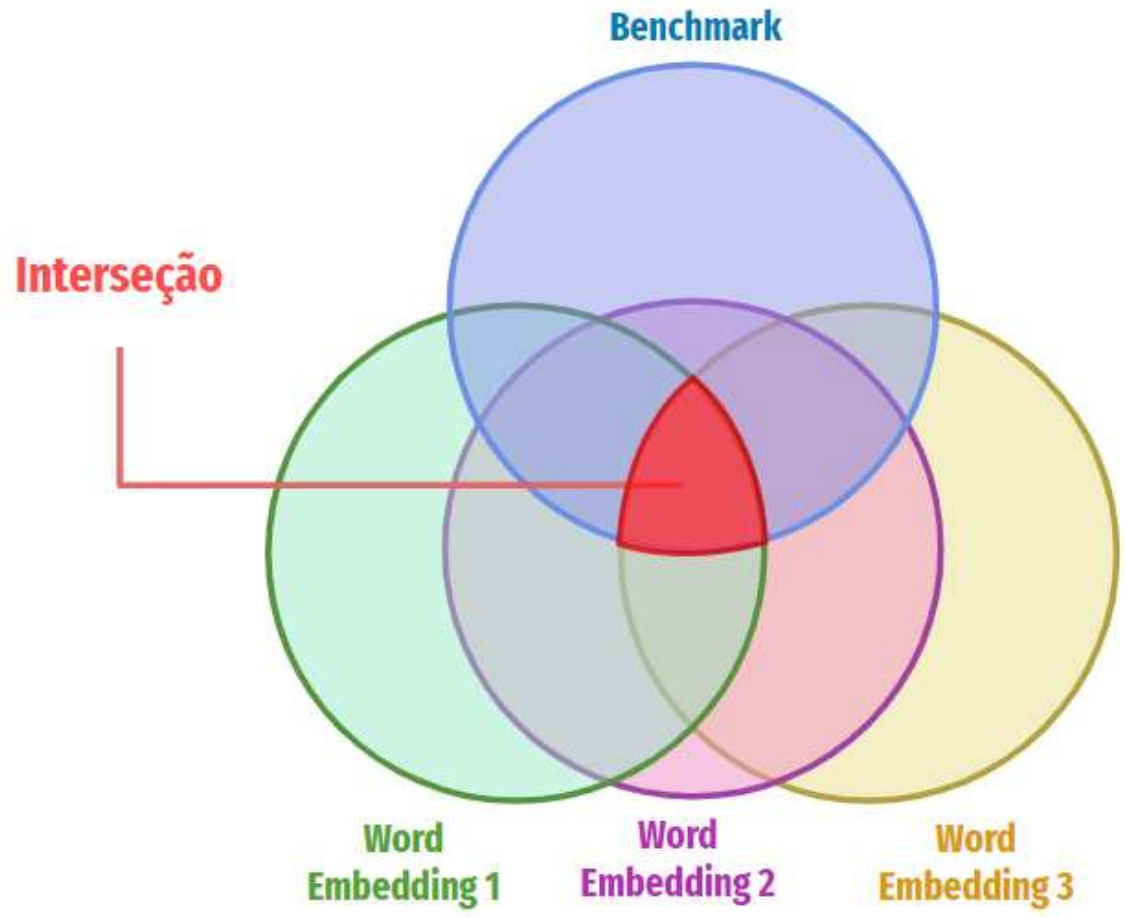
O primeiro passo foi tratar cada *benchmark* para considerar somente os pares com alta pontuação de similaridade. No caso dos conjuntos do *Card660* e *VerbPair130*, filtrou-se os pares que possuíam três ou mais de pontuação de similaridade. Como a pontuação do *SimLex999* possui um alcance de valor diferente, foram filtrados os pares com valor de similaridade maior ou igual a sete (7).

Em seguida, para que a comparação fosse feita dentro do mesmo espaço vetorial de

[4] Disponível em: <https://pilehvar.github.io/card-660/>
[5] Disponível em: <https://fh295.github.io/simlex.html>
[6] Disponível em: <https://www.researchgate.net/publication/257946666_Gold_Standard_130verbpairs>

palavras, o próximo passo foi filtrar as palavras presentes nas *Word Embeddings* pré-treinadas. Para cada *benchmark* filtrado, considerou-se apenas os pares de palavras que existiam ao mesmo tempo nos três espaços vetoriais de cada *embedding*. Criando, assim, um vocabulário de palavras em comum que consiste na interseção das palavras filtradas do *benchmark* com o vocabulário das *Word Embeddings*. A Figura 3 exemplifica visualmente a ideia.

Figura 4 – Interseção entre um *benchmark* e três *Word Embeddings*



Fonte: elaborado pelo autor (2022).

Após o pré-processamento, cada interseção ficou com tamanhos diferentes de vocabulários: o *Card660* ficou com sessenta e quatro (64) palavras únicas, o *VerbPair130* com apenas cinquenta e oito (58), o *SimLex999* com quatrocentas e cinquenta e oito (458). Com exceção do *SimLex999*, os outros dois *benchmarks* possuem poucas palavras que se encontram nas *Word Embeddings*, o *Card660* foi o que mais sofreu redução após essa interseção, foi reduzido em cerca de 95% do seu total de mil trezentas e vinte palavras. A Tabela 2 mostra o total de palavras únicas que restou em cada *benchmark* e uma porcentagem de redução em relação ao seu valor anterior.

Tabela 2 – Vocabulário após interseção

Benchmark	Vocabulário	Redução
Card660	64	95%
VerbPair130	58	77.7%
SimLex999	458	77%

Fonte: elaborado pelo autor (2022).

4.3 Avaliação das *Word Embeddings*

Com os dados tratados, este procedimento consistiu em implementar a avaliação das três *Word Embeddings* em relação ao vocabulário de cada *benchmark*. A avaliação se deu por meio da similaridade entre as palavras, calculando a similaridade entre os vetores de palavras e comparando com a anotação no *benchmark*, dessa forma, se a palavra anotada no *benchmark* como mais similar for a mesma palavra calculada como mais similar no espaço vetorial, é marcado o resultado "True"para aquela palavra.

A métrica usada para fazer o cálculo da semelhança entre os vetores de palavras veio de um cálculo da Álgebra Linear, a similaridade cosseno, que trata-se de uma medida de similaridade entre dois vetores dentro de um espaço vetorial que avalia o valor do cosseno do ângulo compreendido entre eles (Equação 4.1). O resultado desse cálculo é um valor que varia de menos um (-1) a um (1), o que significa que quanto mais próximo de 1 mais similares são os vetores. A maior vantagem dessa medida, é que ela pode ser calculada com vetores de $\boldsymbol{n}$ dimensões, e, como a medida se baseia no ângulo, dois vetores podem ser considerados semelhantes mesmo que estejam em posições diferentes no espaço e/ou possuam grandezas diferentes.

$$similaridade = \cos(\theta) = \frac{A.B}{|A|.|B|} \tag{4.1}$$

Na implementação, considerando cada vocabulário gerado após a interseção dos *benchmarks* com as *Word Embeddings*, foi feito um laço de repetição duplo para cada espaço vetorial das *embeddings*, percorrendo inicialmente todas as palavras do vocabulário e depois calculando a similaridade cosseno entre os vetores de cada palavra dentro do vocabulário, armazenando somente aquela que possui o maior valor de similaridade dentro da intersação, resumidamente, foi armazenada para cada palavra do vocabulário qual a palavra dentro do mesmo vocabulário com valor de similaridade mais próximo de 1.

Por fim, com os dados da palavra mais similar armazenados, através da implementação de mais um laço de repetição, a palavra dita mais similar é comparada com a palavra anotada do *benchmark*, armazenando em uma nova coluna no meu conjunto de dados o valor verdadeiro (*true*) ou falso (*false*) dependendo se a palavra for a mesma ou não, ou seja, se quem está mais próxima no espaço vetorial é a mesma palavra anotada no *benchmark*. A Figura 5 apresenta um excerto de como ficou o conjunto de dados da *Word Embedding* pré-treinada *Word2Vec* sendo avaliada dentro do vocabulário do *Card660*.

Figura 5 – Avaliação *Word2Vec* e *Card660*

	word	top_similar	similarity	target_word	result
0	convocation	convention	0.366433	gathering	False
1	gathering	convention	0.420202	convocation	False
2	care	cognizance	0.250321	caution	False
3	caution	rule	0.217728	care	False
4	prospector	cooperator	0.186309	sourdough	False
5	sourdough	shapeless	0.205783	prospector	False
6	decomposition	pestis	0.274575	factorization	False
7	factorization	amorphous	0.320316	decomposition	False
8	unforeseen	unanticipated	0.775436	unanticipated	True
9	unanticipated	unforeseen	0.775436	unforeseen	True
10	fancifully	whimsically	0.502676	whimsically	True
11	whimsically	fancifully	0.502676	fancifully	True

Fonte: elaborado pelo autor (2022).

4.4 Visualização dos espaços vetoriais

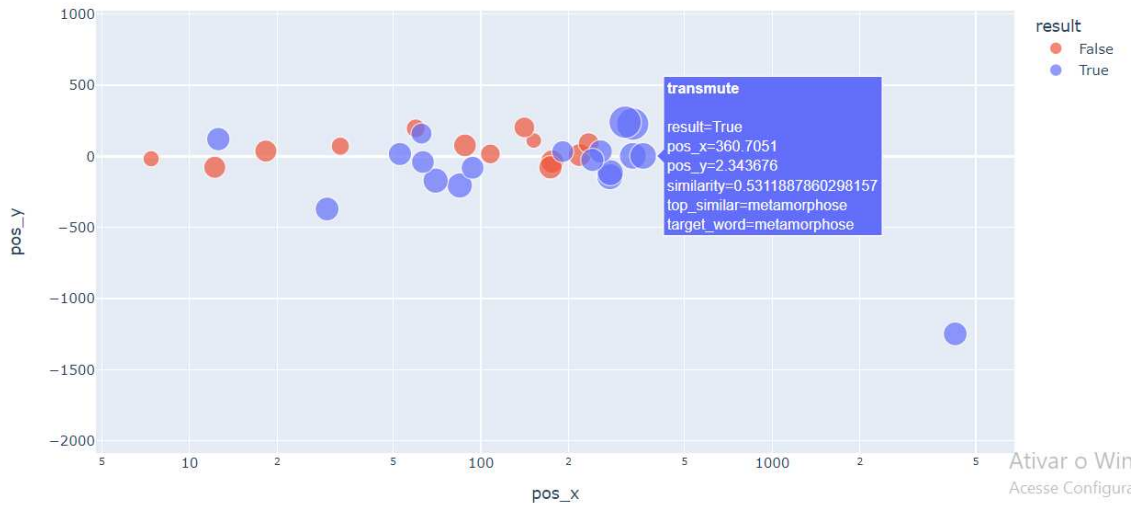
Este processo está dividido em duas partes, no qual, a primeira consiste no redimensionamento dos vetores de palavras usados no vocabulário intersecionado para 2 dimensões, e o segundo na geração de um gráfico de dispersão que possibilita a visualização desses vetores no espaço vetorial das *embeddings*.

Na primeira parte, usou-se uma estratégia de redimensionamento dos vetores, mapeando de trezentas para duas dimensões, com o algoritmo *T-distributed Stochastic Neighbourhood Embedding* (*tSNE*) desenvolvido por Maaten e Hinton (2008). Então, após o redimensionamento, foram armazenadas no conjunto de dados da avaliação de cada *Word Embedding* duas novas colunas para as posições *x* e *y* de cada palavra.

Em trabalhos mais recentes na área, como o *Embedding Comparator*, também é utilizada a estratégia de mapear vetores altamente dimensionais para gerar visualizações interpretáveis para o ser humano (BOGGUST *et al.*, 2022). O motivo do uso dessa estratégia de redimensionamento, é que sem ela seria impossível de visualizar os espaços vetoriais das *Word Embeddings*, pois cada vetor possui trezentas dimensões, o que é incompatível com a visualização espacial dos seres humanos.

A segunda parte do procedimento foi plotar os espaços vetoriais redimensionados em um gráfico de dispersão. Foram considerados os seguintes parâmetros ao gerar o gráfico: as posições *x* e *y* do redimensionamento para plotar os pontos nos eixos *x* e *y* respectivamente; o resultado da avaliação da *Word Embedding* com a palavra anotada (verdadeiro ou falso) foi usado na variação de cor, sendo a cor azul o verdadeiro e a vermelha o falso, ou seja, um ponto azul significa que a palavra com maior similaridade cosseno é igual à palavra anotada dita como mais similar no *benchmark*; o valor de similaridade da palavra com a mais próxima foi usado para o tamanho do ponto no gráfico. A Figura 6 mostra um exemplo de plotagem do espaço vetorial.

Figura 6 – Plotagem *Word2Vec* e *Card660*



Fonte: elaborado pelo autor (2022).

Foi usado também um valor fixo no parâmetro *random state* da função de redimensi-

onamento *tSNE*, para garantir que as plotagens permaneçam sempre as mesmas caso o código seja executado novamente. As plotagens de cada *Word Embedding* com os *benchmarks* serão apresentadas a seguir no capítulo 5.

5 RESULTADOS

Neste capítulo encontram-se os resultados da avaliação proposta e realizada, o capítulo está dividido nas seguintes seções: *Card660*, *VerbPair130* e *SimLex999*. Os *notebooks* gerados para a avaliação deste trabalho estão disponíveis em uma plataforma online própria para repositórios de códigos[1].

5.1 *Card660*

Esta seção apresenta os resultados da avaliação das *Word Embeddings* usando o *benchmark Card660*. Para contextualizar, a Figura 7 mostra uma parte da saída de dados do *Card660* após o filtro dos pares com pontuação de semelhança maior ou igual a 3.

Figura 7 – Pares de palavras *Card660*

	first_word	second_word	score
0	Pokemon	Pocket_Monsters	3.81
5	iight	ok	3.94
6	ACL	EMNLP	3.13
10	full-HD	1080p	4.00
12	convocation	gathering	3.56
14	heater	convector	3.50
17	Hero's_engine	aeolipile	4.00
19	MacBook	ZenBook	3.13
24	Winamp	VLC_media_player	3.19
26	Malva_parviflora	cheeseweed	4.00

Fonte: elaborado pelo autor (2022).

Após obter a interseção das *embeddings Word2Vec*, *FastText* e *GloVe* com as palavras

[1] Disponível em: <https://github.com/ulissessds/tcc-word-embeddings-evaluation>

do *Card660*, ou seja, usando somente o vocabulário do que existia ao mesmo tempo nas três *Word Embeddings*, sobraram apenas sessenta e quatro palavras únicas, especula-se que o motivo seja por conta do *benchmark* ser formado principalmente por palavras infrequentes.

Cada palavra dentro desse vocabulário intersecionado foi testada para checar qual seria a palavra mais próxima dela usando a similaridade cosseno, armazenando tanto o valor obtido no cálculo, como a palavra mais próxima. Depois, comparando a palavra computada com a palavra alvo anotada pelo *benchmark*, foi armazenado, em uma coluna chamada "result", se as palavras eram iguais ou não (verdadeiro ou falso). Esse teste foi feito em cada *Word Embedding*, e cada uma obteve porcentagens de acerto diferentes como mostra a Tabela 3 (os resultados foram arredondados em duas casas decimais para facilitar a leitura da informação).

Tabela 3 – Porcentagem de acerto *Card660*

Word Embedding	Acerto
Word2Vec	53.12%
FastText	68.75%
GloVe	48.44%

Fonte: elaborado pelo autor (2022).

Em seguida, após o redimensionamento dos vetores de cada *embedding*, foram gerados gráficos de dispersão para cada *Word Embedding* usando como espaço vetorial apenas as palavras que estavam na interseção das três embeddings com o *benchmark Card660*. Os parâmetros para a construção do gráfico foram baseados nos dados gerados pela avaliação, sendo a cor azul as palavras que acertaram a similaridade com a anotação do *benchmark*, e a cor vermelha as que erraram, o tamanho de cada ponto é dado pelo valor de similaridade cosseno da palavra em questão com a sua palavra mais próxima.

As plotagens geradas envolvendo o *Card660* encontram-se no Apêndice A. Como o espaço vetorial é muito pequeno, percebe-se que os dados não foram suficientes para geração de aglomerações notáveis entre as palavras, e, em todas as *embeddings*, as palavras parecem ter uma distribuição bem proporcional independentemente da palavra ter acertado ou não a sua palavra mais similar.

5.2 *Verbpair130*

Esta seção apresenta os resultados da avaliação das *Word Embeddings* usando o *benchmark VerbPair130*. A mesma lógica de pré-processamento que foi usada no *Card660*

também foi usada no *VerbPair130*, pois ambos possuem uma variação igual de zero a quatro para a pontuação de similaridade. A Figura 8 mostra uma pequena fração do conjunto de dados do *VerbPair130*.

Figura 8 – Pares de palavras *VerbPair130*

	first_word	second_word	score
0	brag	boast	4.000
1	concoct	devise	4.000
2	divide	split	4.000
3	build	construct	4.000
4	end	terminate	4.000
5	accentuate	highlight	4.000
6	demonstrate	show	3.833
7	solve	figure out	3.833
8	consume	eat	3.833
9	position	situate	3.833

Fonte: elaborado pelo autor (2022).

Após a interseção do vocabulário do *VerbPair130* com as *Word Embeddings*, o total de palavras reduziu para cinquenta e oito, e, em seguida, para implementar a avaliação das *embeddings*, a mesma lógica usada no *notebook* do *Card660* também foi usada para o *VerbPair130*, ou seja, foi calculada a similaridade entre as palavras e depois comparada com os dados anotados do *benchmark*. A Tabela 4 mostra a taxa de acerto para cada *embedding* após a avaliação.

Contudo, mesmo obtendo uma performance diferente de acertos, as plotagens dos espaços vetoriais com o *VerbPair130* não foram tão diferentes das geradas com o *Card660*, possuindo também uma dispersão balanceada e sem relação aparente no gráfico entre os resultados de acertos e erros (visualizações no Apêndice B).

Tabela 4 – Porcentagem de acerto *VerbPair130*

Word Embedding	Acerto
Word2Vec	46.55%
FastText	53.45%
GloVe	39.65%

Fonte: elaborado pelo autor (2022).

5.3 *SimLex999*

A seguir, serão apresentados os resultados da avaliação das *Word Embeddings* usando o *benchmark SimLex999*. A primeira etapa de pré-processamento para o *SimLex999* teve um parâmetro diferente, pois a pontuação de similaridade dos dados varia de zero a dez, então, o filtro foi aplicado para as palavras com pontuação maior ou igual a sete. A Figura 9 mostra um excerto do conjunto de dados do *SimLex999*.

Figura 9 – Pares de palavras *SimLex999*

	first_word	second_word	score
1	smart	intelligent	9.20
2	hard	difficult	8.77
3	happy	cheerful	9.55
5	fast	rapid	8.75
6	happy	glad	9.17
8	stupid	dumb	9.58
9	weird	strange	8.93
11	bad	awful	8.42
13	bad	terrible	7.78
16	insane	crazy	9.57

Fonte: elaborado pelo autor (2022).

O mesmo procedimento de interseção e avaliação utilizados nos *notebooks* dos

benchmarks anteriores foi usado também com o *SimLex999*. Entretanto, possivelmente por ser o maior conjunto de dados, com um total de mil novecentas e noventa e oito palavras, o seu vocabulário foi o maior após a interseção, ficando com quatrocentas e cinquenta e oito palavras únicas.

A performance da avaliação das *embeddings* usando o *SimLex999* para acertar a palavra mais similar não foi tão diferente das anteriores, para todas as *embeddings* a taxa de acerto foi menor do que a metade (vide Tabela 5).

Tabela 5 – Porcentagem de acerto *SimLex999*

Word Embedding	Acerto
Word2Vec	45.41%
FastText	49.34%
GloVe	41.05%

Fonte: elaborado pelo autor (2022).

O Apêndice C mostra as visualizações geradas dos espaços vetoriais das *embeddings* usando o *SimLex999*. Diferentemente dos outros *benchmarks*, por conta do vocabulário mais extenso, as plotagens são mais rica em dados. Porém, mesmo com mais palavras, não foi possível identificar aglomerações específicas e os dados continuam distribuídos de maneira balanceada no espaço.

O capítulo seguinte traz as conclusões advindas dos resultados aqui apresentados e também discute ideias para trabalhos futuros.

6 CONCLUSÕES E TRABALHOS FUTUROS

A avaliação intrínseca realizada nas *Word Embeddings* pré-treinadas resultou em dados consistentes, apesar do tamanho diferente de cada *benchmark*. Como pode ser visto na Tabela 6, é possível estabelecer um ranqueamento da performance de cada *Word Embedding* em relação aos *benchmarks* utilizados. A performance do *FastText* foi a melhor em todos os cenários, enquanto as *embeddings Word2Vec* e *GloVe* ficaram em segundo e terceiro lugar, respectivamente.

Tabela 6 – Ranqueamento das *Word Embeddings*

	Word2Vec	*Fasttext*	*GloVe*
Card660	53.12%	68.75%	48.44%
VerbPair130	46.55%	53.45%	39.65%
SimLex999	45.41%	49.34%	41.05%

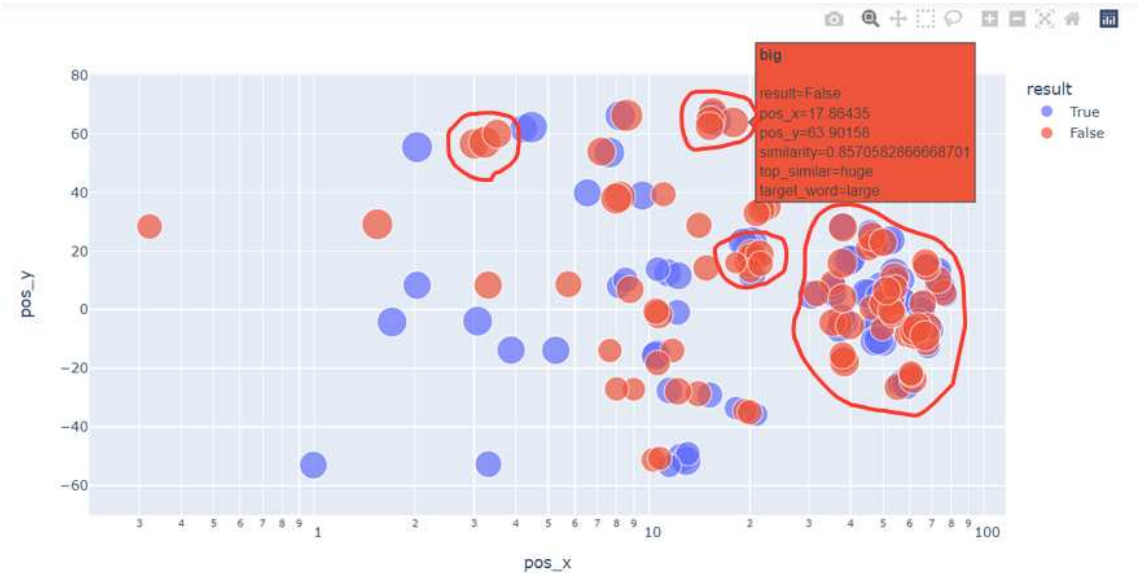
Fonte: elaborado pelo autor (2022).

Sobre as plotagens (vide Apêndices A, B e C), as palavras estão todas bem distribuídas no espaço vetorial, independentemente dos *benchmarks*, de forma que não são geradas aglomerações significativas para observar padrões com base nas distâncias. Olhando por outro ponto de análise, observa-se também que a quantidade de pontos vermelhos (erros) é visualmente parecida com a de pontos azuis (acertos), isso se dá pela baixa performance de acertos, como visto na Tabela 6.

Contudo, analisando mais a fundo a semântica das palavras dos pontos vermelhos, percebe-se que a palavra aprendida pela *embedding* como a mais similar mantém uma relação semântica com o par de palavras do ponto. A Figura 10 mostra um exemplo disso com palavras da plotagem do *GloVe* com o *SimLex999*. Nesse caso, a palavra analisada é "*Big*"e a *Word Embedding* aprendeu como mais similar a palavra "*Huge*", no entanto, a palavra alvo era "*Large*", e todas as 3 possuem significados semânticos parecidos, traduzindo do inglês podem ser entendidas como "Grande". O que leva a conclusão que olhar somente para a palavra mais próxima não seja o ideal para esse tipo de avaliação de similaridade entre palavras, pois, normalmente, uma mesma palavra possui uma série de sinônimos.

O desenvolvimento de um modelo de Aprendizagem Profunda para resolver uma tarefa de PLN é um processo que demanda muitas decisões por parte dos cientistas de dados. E, com os resultados obtidos na avaliação deste trabalho, nota-se como é difícil escolher qual *Word Embedding* pré-treinada enxaica melhor no contexto do seu problema. Todavia, presume-se que os resultados da avaliação poderiam ser diferentes caso as *embeddings*, usando os mesmos

Figura 10 – Exemplo de erro de similaridade



Fonte: elaborado pelo autor (2022).

algoritmos, fossem treinadas em outro contexto, ou fossem usados *benchmarks* diferentes na avaliação.

Como um trabalho futuro, caberia um possível questionamento para esse mesmo tipo de avaliação com *embeddings* pré-treinadas em contextos diferentes. Por exemplo, se houvessem *benchmarks* com palavras em português, poderia-se realizar o mesmo tipo de avaliação para confirmar os mesmos resultados em outro idioma. O que leva a outra possibilidade investigativa futura, uma proposta de criação desses *benchmarks* no idioma português, pois já existem *Word Embeddings* pré-treinadas em contextos genéricos (HARTMANN *et al.*, 2017b).

Além disso, possíveis trabalhos futuros poderiam também realizar a comparação extrínseca (comparação dos modelos de *Word Embeddings* em diferentes tarefas de PLN a partir do treinamento da rede neural), e também realizar a comparação intrínseca e extrínseca de *embeddings* de sentença, os *Sentence Embeddings* comentados no Capítulo 2.

Este trabalho foi sobretudo investigativo, nos deixando com um experimento válido dentro da área que pode ser facilmente replicado usando outras *embeddings* e *benchmarks*, dando margem para mais questionamentos e possibilidades futuras de investigação. Sendo assim, cito aqui algumas ações que poderiam servir para continuar esse tipo de avaliação:

1. Usar dados em português, *benchmarks* e *embeddings* pré-treinadas.
2. Utilizar outras métricas para a comparação de similaridade, por exemplo, comparar com as cinco mais similares.
3. Avaliar uma *embedding* de cada vez, sem interseção dos espaços vetoriais.

4. Construir um *benchmark* no idioma português com um número considerável de pares de palavras similares.

Desta forma, este trabalho conseguiu trazer uma maior compreensão sobre *Word Embeddings* e métodos de avaliação, reafirmando a suspeita inicial sobre a grande dificuldade envolvida na escolha de qual *embedding* usar ao desenvolver um modelo para tarefas de *PLN*.

REFERÊNCIAS

BENGIO, Y.; DUCHARME, R.; VINCENT, P. A neural probabilistic language model. **Advances in neural information processing systems**, v. 13, 2000.

BOGGUST, A.; CARTER, B.; SATYANARAYAN, A. Embedding Comparator: Visualizing Differences in Global Structure and Local Neighborhoods via Small Multiples. In: **ACM Intelligent User Interfaces (IUI)**. [s.n.], 2022. Disponível em: <http://vis.csail.mit.edu/pubs/embedding-comparator>.

BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. Enriching word vectors with subword information. **Transactions of the association for computational linguistics**, MIT Press, v. 5, p. 135–146, 2017.

CER, D.; YANG, Y.; KONG, S.-y.; HUA, N.; LIMTIACO, N.; JOHN, R. S.; CONSTANT, N.; GUAJARDO-CESPEDES, M.; YUAN, S.; TAR, C. *et al.* Universal sentence encoder. **arXiv preprint arXiv:1803.11175**, 2018.

CHIDAMBARAM, M.; YANG, Y.; CER, D.; YUAN, S.; SUNG, Y.-H.; STROPE, B.; KURZWEIL, R. Learning cross-lingual sentence representations via a multi-task dual-encoder model. **arXiv preprint arXiv:1810.12836**, 2018.

CHOLLET, F. **Deep learning with Python**. [S.l.]: Manning Publications, 2018.

FINKELSTEIN, L.; GABRILOVICH, E.; MATIAS, Y.; RIVLIN, E.; SOLAN, Z.; WOLFMAN, G.; RUPPIN, E. Placing search in context: The concept revisited. In: **Proceedings of the 10th international conference on World Wide Web**. [S.l.: s.n.], 2001. p. 406–414.

FIRMIANO, A.; SILVA, T. L. C. D. Identifying duplicate police reports. In: IEEE. **2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)**. [S.l.], 2021. p. 244–247.

GRAY, J. **Database and Transaction Processing Performance Handbook.** 1993.

HARTMANN, N.; FONSECA, E.; SHULBY, C.; TREVISO, M.; RODRIGUES, J.; ALUISIO, S. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. **arXiv preprint arXiv:1708.06025**, 2017.

HARTMANN, N.; FONSECA, E.; SHULBY, C.; TREVISO, M.; SILVA, J. da; ALUISIO, S. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. 08 2017.

HILL, F.; REICHART, R.; KORHONEN, A. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. **Computational Linguistics**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 41, n. 4, p. 665–695, 2015.

INOHARA, K.; UTSUMI, A. Jwsan: Japanese word similarity and association norm. **Language Resources and Evaluation**, Springer, v. 56, n. 1, p. 109–137, 2022.

JÚNIOR, V. O. D. S.; BRANCO, J. A. C.; OLIVEIRA, M. A. D.; SILVA, T. L. C. D.; CRUZ, L. A.; MAGALHAES, R. P. A natural language understanding model covid-19 based for chatbots. In: IEEE. **2021 IEEE 21st International conference on bioinformatics and bioengineering (BIBE)**. [S.l.], 2021. p. 1–7.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: PMLR. **International conference on machine learning**. [S.l.], 2014. p. 1188–1196.

MAATEN, L. van der; HINTON, G. Visualizing data using t-sne. **Journal of Machine Learning Research**, v. 9, n. 86, p. 2579–2605, 2008. Disponível em: <http://jmlr.org/papers/v9/vandermaaten08a.html>.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed representations of words and phrases and their compositionality. **Advances in neural information processing systems**, v. 26, 2013.

OLIVEIRA, B. S. N. Aprendizado profundo para reconhecimento de entidades nomeadas em narrativas de roubos. 2020.

OTTER, D. W.; MEDINA, J. R.; KALITA, J. K. A survey of the usages of deep learning for natural language processing. **IEEE transactions on neural networks and learning systems**, IEEE, v. 32, n. 2, p. 604–624, 2020.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1532–1543.

PILEHVAR, M. T.; KARTSAKLIS, D.; PROKHOROV, V.; COLLIER, N. Card-660: Cambridge rare word dataset-a reliable benchmark for infrequent word representation models. **arXiv preprint arXiv:1808.09308**, 2018.

QI, Y.; SACHAN, D. S.; FELIX, M.; PADMANABHAN, S. J.; NEUBIG, G. When and why are pre-trained word embeddings useful for neural machine translation? **arXiv preprint arXiv:1804.06323**, 2018.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. **arXiv preprint arXiv:1908.10084**, 2019.

SCHNABEL, T.; LABUTOV, I.; MIMNO, D.; JOACHIMS, T. Evaluation methods for unsupervised word embeddings. In: **Proceedings of the 2015 conference on empirical methods in natural language processing**. [S.l.: s.n.], 2015. p. 298–307.

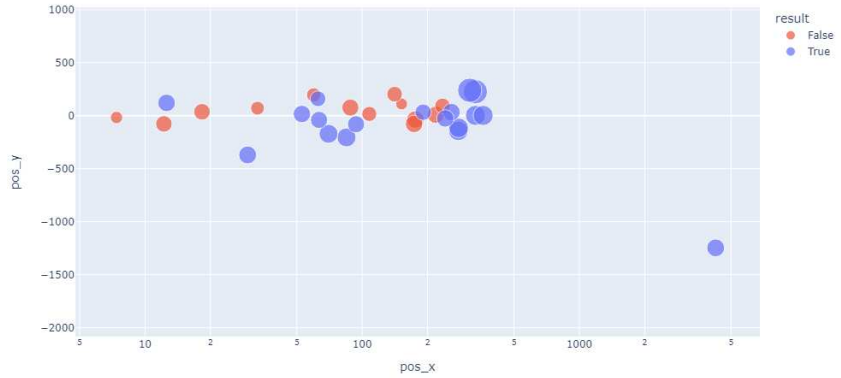
TORREGROSSA, F.; ALLESIARDO, R.; CLAVEAU, V.; KOOLI, N.; GRAVIER, G. A survey on training and evaluation of word embeddings. **International Journal of Data Science and Analytics**, Springer, v. 11, n. 2, p. 85–103, 2021.

TOSHEVSKA, M.; STOJANOVSKA, F.; KALAJDJIESKI, J. Comparative analysis of word embeddings for capturing word similarities. **arXiv preprint arXiv:2005.03812**, 2020.

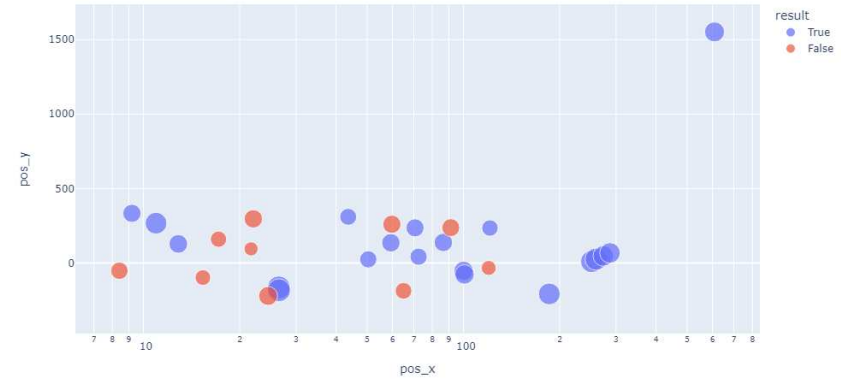
YANG, D.; POWERS, D. **130 verb pairs for lexical similarity test**. 2015.

APÊNDICE A – PLOTAGENS *CARD660*

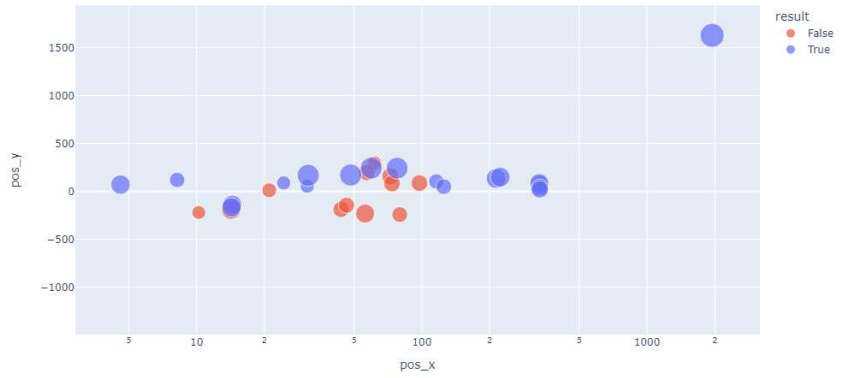
Card660 e *Word2Vec*



Card660 e *FastText*

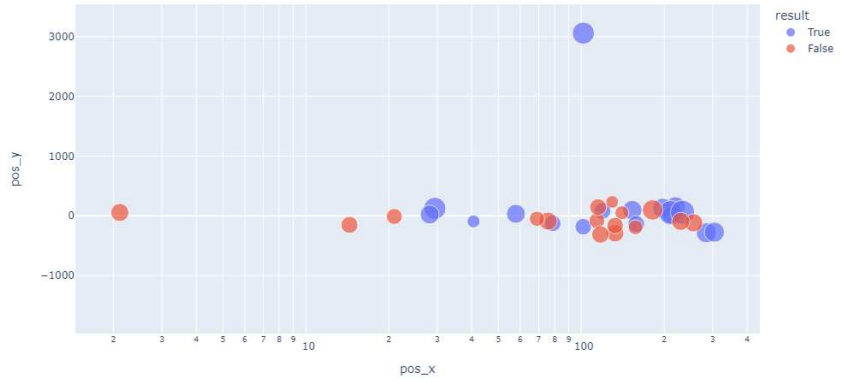


Card660 e *GloVe*

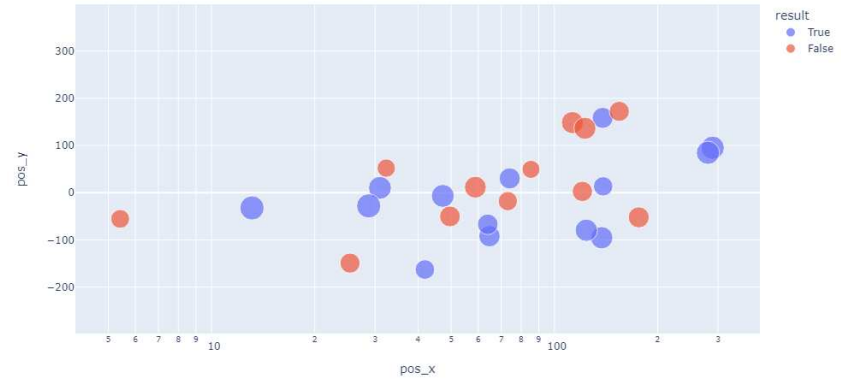


APÊNDICE B – PLOTAGENS *VERBPAIR130*

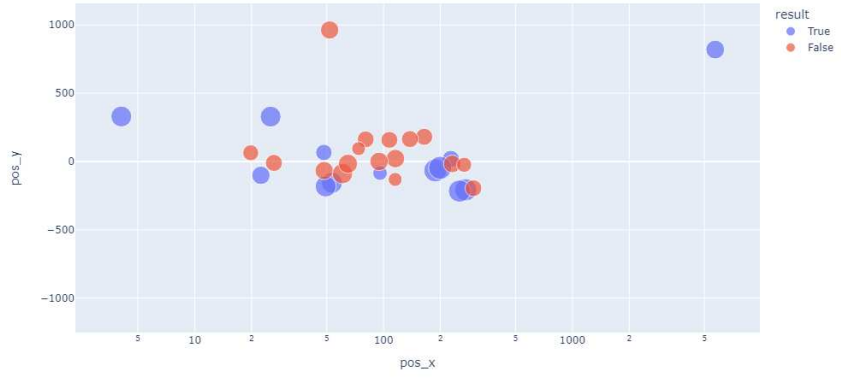
VerbPair130 e *Word2Vec*



VerbPair130 e *FastText*

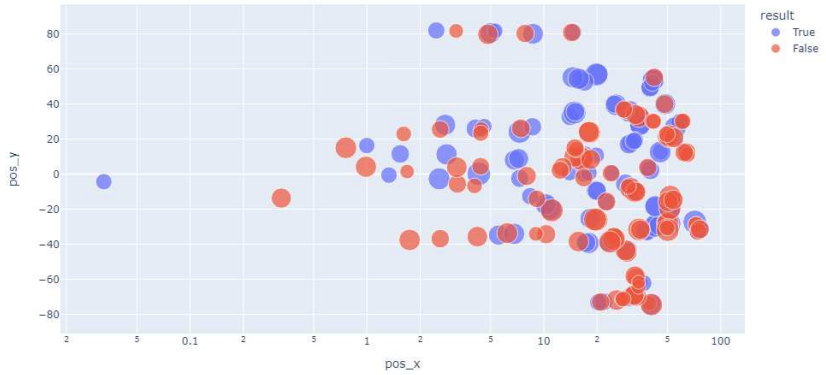


VerbPair130 e *GloVe*

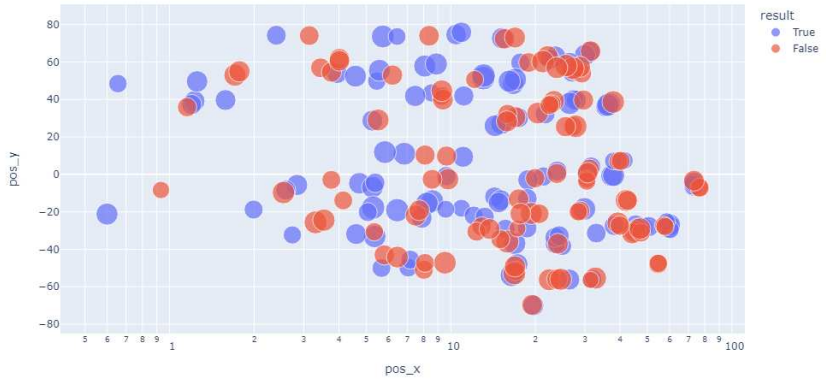


APÊNDICE C – PLOTAGENS *SIMLEX999*

SimLex999 e *Word2Vec*



SimLex999 e *FastText*



SimLex999 e *GloVe*