Pontifícia Universidade Católica
do Rio de Janeiro

Ivan de Jesus Pereira Pinto

Corpus para o Domínio Acadêmico: Modelos e Aplicações

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática, do Departamento de Informática da PUC-Rio.

Orientador: Prof. Sérgio Colcher

Rio de Janeiro
Setembro de 2021

Pontifícia Universidade Católica
do Rio de Janeiro

Ivan de Jesus Pereira Pinto

Corpus para o Domínio Acadêmico: Modelos e Aplicações

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós–graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo:

Prof. Sérgio Colcher
Orientador
Departamento de Informática – PUC-Rio

Prof. Jônatas Wehrmann
Departamento de Informática – PUC-Rio

Prof. Julio Cesar Duarte
Instituto Militar de Engenharia – IME

Rio de Janeiro, 15 de Setembro de 2021

Ivan de Jesus Pereira Pinto

Graduou-se em Ciência da Computação (2018) pela Universidade Federal do Maranhão - Campus Dom Delgado (UFMA)

Ficha Catalográfica

Pereira Pinto, Ivan de Jesus

Corpus para o Domínio Acadêmico: Modelos e Aplicações / Ivan de Jesus Pereira Pinto; orientador: Sérgio Colcher. – 2021.

84 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2021.

Inclui bibliografia

1. Processamento de Linguagem Natural – Teses. 2. Aprendizado de Máquina – Teses. 3. Embeddings – Teses. 4. Pergunta-Resposta Interativo. – Teses. 5. Processamento de Linguagem Natural. 6. Aprendizado de Máquina. 7. Embeddings. 8. Pergunta-Resposta Interativo.. I. Colcher, Sérgio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

Primeiramente, agradeço à PUC-Rio pelo ambiente de estudo oferecido aos seus alunos. Agradeço ao laboratório que faço parte, o *TELEMÍDIA*, por todo suporte pessoal e de recursos nesses dois anos.

Agradeço ao *Google TPU Research Cloud* pelas *GPUs* e *TPUs* disponibilizadas, essenciais para essa pesquisa. Agradeço aos meus pais e as minhas tias Célida e Selma, que me ajudaram no caminho até a PUC-Rio. Um agradecimento enorme a Jéssica Paloma, pela ajuda na edição desta dissertação e no desenvolvimento das aplicações.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Por fim, gostaria de agradecer ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo financiamento deste trabalho.

Resumo

Pereira Pinto, Ivan de Jesus; Colcher, Sérgio. **Corpus para o Domínio Acadêmico: Modelos e Aplicações**. Rio de Janeiro, 2021. 73p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Dados acadêmicos (e.g., Teses, Dissertações) englobam aspectos de toda uma sociedade, bem como seu conhecimento científico. Neles, há uma riqueza de informações a ser explorada por modelos computacionais, e que podem ser positivos para sociedade. Os modelos de aprendizado de máquina, em especial, possuem uma crescente necessidade de dados para treinamento, que precisam ser estruturados e de tamanho considerável. Seu uso na área de processamento de linguagem natural é pervasivo nas mais diversas tarefas.

Este trabalho realiza o esforço de coleta, construção, análise do maior corpus acadêmico conhecido na língua portuguesa. Foram treinados modelos de vetores de palavras, *bag-of-words* e *transformer*. O modelo *transformer* BERTAcadêmico apresentou os melhores resultados, com 77% de f1-score na classificação da Grande Área de conhecimento e 63% de f1-score na classificação da Área de conhecimento nas categorizações de Teses e Dissertações.

É feita ainda uma análise semântica do corpus acadêmico através da modelagem de tópicos, e uma visualização inédita das áreas de conhecimento em forma de *clusters*. Por fim, é apresentada uma aplicação que faz uso dos modelos treinados, o SucupiraBot.

Palavras-chave

Processamento de Linguagem Natural; Aprendizado de Máquina; Embeddings; Pergunta-Resposta Interativo.

Abstract

Pereira Pinto, Ivan de Jesus; Colcher, Sérgio (Advisor). **Corpus for Academic Domain: Models and Applications**. Rio de Janeiro, 2021. 73p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Academic data (i.e., Thesis, Dissertation) encompasses aspects of a whole society, as well as its scientific knowledge. There is a wealth of information to be explored by computational models, and that can be positive for society. Machine learning models in particular, have an increasing need for training data, that are efficient and of considerable size. Its use in the area of natural language processing (NLP) is pervasive in many different tasks.

This work makes the effort of collecting, constructing, analyzing and training of models for the biggest known academic corpus in the Portuguese language. Word embeddings, bag of words and transformers models have been trained. The Bert-Academico has shown the better result, with 77% of f1-score in Great area of knowledge and 63% in knowledge area classification of Thesis and Dissertation.

A semantic analysis of the academic corpus is made through topic modelling, and an unprecedented visualization of the knowledge areas is presented. Lastly, an application that uses the trained models is showcased, the SucupiraBot.

Keywords

Neural Language Processing; Machine Learning; Embeddings; Interactive Question Answer.

Sumário

Lista de figuras

Lista de tabelas

Quanto mais se planeja cuidadosamente, mais eventos inesperados acontecem.

Dio Brando, *JoJo's Bizarre Adventure: Phantom Blood.*

1
Introdução

A área de processamento de linguagem natural (PLN) tem evoluído muito nos últimos anos. Seu objetivo geral é permitir a compreensão e reprodução de textos em linguagem natural, permitindo a realização de tarefas como a redação de sentenças ou a interpretação textos. Sua evolução está associada aos campos emergentes de aprendizado de máquina e estatístico, que fazem uso do aumento cada vez mais expressivo na disponibilidade de dados utilizados na construção de coleções de documentos, também denominados *corpus*. Modelos são desenvolvidos para solucionar problemas recorrentes na PLN, como reconhecimento de entidade nomeada (NER - *Named entity recognition*), extração de relação (RE - *Relation Extraction*), dentre outros.

Mais recentemente, modelos de redes neurais têm sido frequentemente desenvolvidos na área [5] [79] [80]. Diferentemente dos métodos estatísticos, tais modelos não requerem o uso elaborado de engenharia de características. As redes neurais são capazes de aprender características significantes de modo automático, bastando terem a sua disposição um número relevante de dados. Uma técnica popular onde os modelos neurais tem obtido bastante sucesso são os *word embeddings*, úteis para aprender a semântica de palavras e sentenças, ao representar esses elementos em um espaço vetorial no qual representações próximas podem indicar que os termos são semanticamente semelhantes.

O desenvolvimento de modelos estatísticos e neurais em português tem uma historia recente, sendo significantemente guiada pelos tradicionais congressos e competições como a *International Conference on the Computational Processing of Portuguese* (PROPOR). É importante não esquecer, no entanto, que o português possui duas variantes ortográficas: a europeia e a brasileira. Apesar de terem uma grande interseção, modelos especializados em uma podem não ter o mesmo resultado na outra. Além delas, o português falado em países lusófonos como Moçambique, Angola, e outros podem apresentar pequenas variações.

Modelos e corpora para o português falado coloquialmente têm sido desenvolvidos pelo esforço de se coletar dados de fontes como blogs, redes sociais, e arquivos da Internet. Comparativamente, no entanto, pouco esforço tem sido direcionado para coletar documentos científicos em quantidade significante.

Modelos que possuam conhecimento de domínio são vitais para diversas tarefas e aplicações. Buscadores, ontologias, categorizadores de *email* e *spam*, sistemas pergunta e resposta, dentre outros, podem possuir a maior eficácia possível se o conhecimento sintático e semântico com os quais foram treinados ou desenvolvidos não tiver sido o suficiente para cobrir seus domínios de interesse. Isso, por si só, já justificaria, a produção de um *corpus* científica; mas podem haver ainda outras aplicações, já que a organização desses corpora pode ser relevante para diversas áreas como biblioteconomia, letras e computação, por exemplo.

Esta dissertação consiste de um esforço de coleta e produção de um *corpus* com conteúdo acadêmico na língua portuguesa brasileira, com suas informações estatísticas de relevância, e mecanismo de visualização dos dados. Propõe-se também realizar o treinamento de modelos, como os de vetores de palavras, e avaliações em tarefas específicas como classificação, NER e RE. Por fim, propõe-se apresentar um caso real de aplicação que faça uso dos modelos: um *bot* pergunta-resposta sobre dados acadêmicos.

O restante deste capítulo está organizado da seguinte forma: são apresentadas as principais motivações e objetivos por detrás do desenvolvimento desta pesquisa, bem como uma visão geral sobre a metodologia ser aplicada e os tópicos de interesses a serem abordados com mais detalhes.

1.1 Motivação

Com o intuito de permitir que as informações sobre os programas de pós graduação estivessem disponíveis ao público, além de tornar transparentes as avaliações realizadas nas pós-graduações, foi criada a Plataforma Sucupira pela CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior). Esse sistema nasceu com a meta de proporcionar maior transparência sobre todos os processos e produções realizadas pelos programas de pós graduação, além de tornar essas informações mais facilmente recuperáveis pelo público.[1]

No Sucupira, há diversas informações acessíveis ao público, tais como lista de discentes, linhas de pesquisa, produção intelectual, trabalhos de conclusão etc. Grande parte das informações da produção gerada pelos programas de pós-graduação brasileiros pode ser obtida por meio dessa ferramenta, se estabelecendo assim como uma fonte confiável de dados abertos.

Dentre os dados disponibilizados pela plataforma, um dos mais relevantes é o da produção intelectual de teses e dissertações. Tais informações compõem

[1] `https://www.gov.br/capes/pt-br/assuntos/noticias/capes-lanca-plataforma-sucupira-para-gestao-da-pos-graduacao`

uma das fontes mais expressivas do conteúdo científico/acadêmico brasileiro, com documentos catalogados e armazenados desde a década de 1980. Uma parcela desses documentos está disponível publicamente via PDFs, de forma não estruturada, sendo necessário o uso de ferramentas para extração do conteúdo textual.

1.2 Objetivos e Contribuições

O objetivo principal dessa dissertação é a construção do *corpus* acadêmico e exploração do seu conhecimento por meio de aplicações propostas. Mais especificamente, são almejados os seguintes objetivos:

- Desenvolvimento de uma metodologia para construção do *corpus* com conteúdo acadêmico/científico expressivo em língua portuguesa
- Extração de informações estatísticas do *corpus*
- Treinamento de modelos de linguagem no *corpus*
- Avaliação dos modelos treinados
- Visualização e extração de conhecimento semântico do *corpus*
- Desenvolvimento de um caso real de aplicação que faça uso dos dados/modelos.

A contribuição do *corpus* é de riqueza difícil de se mensurar, já que cada trabalho possui mais do que simples propostas de técnicas inovadoras ou relatos promissores de experimentos. De fato, pode-se argumentar que muitos desses trabalhos contém, de forma implícita ou explicita, anos de trabalho dedicados (1 ou mais de Mestrados e 4 ou mais de Doutorados) e diversas histórias de vida.

É importante destacar que esses dados e modelos têm relevância também para a comunidade científica no geral, não apenas a nacional. Existe uma gama de dados inerentes às regiões e às realidades em que vivem os autores das teses e dissertações que podem ser úteis em diversos contextos. Os dados vão conter essas informações e os modelos vão (até certo ponto) absorver esse conhecimento, permitindo diversas formas de uso e exploração de conhecimento específico, a exemplo de informações sobre:

1. aspectos da flora e fauna local presente em cada região do país.
2. as linguagens e dialetos indígenas e de povos remotos.
3. as diferentes realidades urbanas e rurais do Brasil

4. as desigualdades sociais e raciais
5. as mazelas sociais (fome, analfabetismo, etc)
6. as diversas culturas do brasil (de origem africana, europeia, asiática..)
7. a culinária brasileira
8. as doenças virais e seus impactos na população (dengue, zika etc)
9. a educação pública, em particular a brasileira.

1.3 Metodologia

Para este trabalho, é proposta uma abordagem para construção do *corpus* acadêmico, descrevendo a coleta dos dados ao uso de ferramenta estado da arte para extração de texto acadêmico, e a posterior limpeza dos dados. Uma vez processados, estatísticas relevantes são calculadas para melhor entendimento do *corpus*, bem como é feita a comparação com outros *corpora* existentes.

São treinados modelos estatísticos básicos como o *TF-IDF*, bem como os modelos de vetores de palavras *word2vec*, *fastText* e *glove*, e o modelo de linguagem contextual BERT. Avaliações dos vetores de palavras e dos modelos em tarefas de classificação são propostas.

A exploração do *corpus* é realizada por meio da modelagem de tópicos e visualização dos vetores de palavras. São utilizadas as técnicas de *Latent Dirichlet Allocation* (*LDA*) e *Uniform Manifold Approximation and Projection* (UMAP) respectivamente.

Por fim, uma proposta de aplicação que faz uso dos modelos é apresentada, o SucupiraBot.

1.4 Organização deste Documento

O restante deste documento está organizado da seguinte maneira: o Capítulo 2 introduz os conceitos necessários para entendimento do trabalho. Em seguida, o Capítulo 3 apresenta um levantamento dos trabalhos mais importantes encontrados na literatura e que se relacionam com o objeto desta dissertação. O Capítulo 4 apresenta a solução proposta, que é a descrição do processo de construção do corpus de conteúdo acadêmico/científico, bem como estatísticas do mesmo. O Capítulo 5 apresenta o treinamento de modelos de linguagem no corpus, e a avaliação em tarefas de analogia e classificação. O Capítulo 6 explora o corpus por meio da visualização de clusterização e modelagem de

tópicos. O Capítulo 7 avalia uma aplicação de caso real dos modelos treinados. O Capítulo 8 faz os comentários finais sobre o trabalho.

2
Fundamentação

Este capítulo apresenta os tópicos teóricos essenciais que serão abordados no trabalho. Os conceitos básicos de tokenização de sub-palavras e vetores de palavras são introduzidos inicialmente pela sua importância nos modelos que serão treinados posteriormente. Em seguida, são apresentados os modelos clássicos de PLN para representação de texto em formatos adequados para os classificadores. A seção seguinte dá o referencial teórico para todos os modelos baseados em redes neurais, variando da mais simples rede totalmente conectada aos *transformers*. Adiante, adentra-se no tópico de Modelos de Linguagem Estáticos, que são aqueles que estão mais profundamente ligados ao treinamento dos vetores de palavras. Já o os Modelos de Linguagem Contextuais enriquecem os vetores de palavras considerando o contexto, e é o caso dos modelos mais atuais como os *transformers*.

Se distanciando mais dos modelos de aprendizado de máquina, mas ainda na área de PLN, aborda-se as tarefas de Modelagem de Tópicos e Redução de dimensionalidade, ambas essenciais em capítulos posteriores para analise semântica e visual do corpus. Por fim são apresentadas algumas das tarefas clássicas de PLN como NER e *Question Answer*, que serão abordadas neste trabalho.

2.1
Tokenização de sub-palavras

O processo de *tokenização* trata de segmentar sentenças em unidades que são consideradas importantes para a aplicação. Até recentemente, a abordagem mais comum era a simples divisão de palavras por *espaço em branco*. No entanto, trabalhos como *fastText*[40] e BERT [20] popularizaram o uso de algoritmos que tokenizam ao nível de sub-palavas, permitindo se ter um vocabulário menor fazendo o reuso de termos, e uma maior capacidade de lidar com palavras fora de vocabulário. Alguns dos métodos mais populares de *tokenização* de sub-palavras são o BPE [44] e o *word-piece*[43], o último o qual será utilizado no treinamento dos modelos deste trabalho.

2.1.1 Wordpiece

O Wordpiece[43] é um algoritmo de segmentação de palavras em sub-palavras, o que dá a vantagem de se trabalhar em um subconjunto menor que o vocabulário, além de lidar melhor com palavras raras e as novas que estão fora do vocabulário, já que todas as palavras vão ser quebradas em sub-palavras conhecidas. O algoritmo pode ser treinado de forma não-supervisionada em uma amostra do corpus de interesse, caso seja muito grande. Ele é descrito nos seguintes passos:

1. Inicializa-se o vocabulário com todos os caracteres presentes no corpus
2. Treina-se um modelo de linguagem no corpus com o vocabulário atual
3. Gera-se novas unidades de palavras ao combinar duas unidades de todo vocabulário atual. Dentre elas, escolhe-se aquela nova unidade que melhora mais a performance do modelo ao ser adicionada no corpus.
4. Volte para 2 até que um número máximo de palavras pré-definido seja alcançado.

2.2 Vetores de palavras

Vetores de palavras são representações em forma de vetores multidimensionais reais das sentenças, localizadas em um espaço vetorial pré-definido, no qual a geometria e a relação entre esses vetores capturam as relações semânticas entre as palavras correspondentes. Eles são geralmente utilizados em conjunto com a similaridade de cosseno, dada pela Equação 2-1.

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\left\|\vec{b}\right\|} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}} \tag{2-1}$$

na qual a e b são os vetores de vetores de palavras. Com esse cálculo de similaridade, palavras com contexto similar ocupam posições mais próximas no espaço. Por exemplo, a similaridade entre palavras como “homem” e “mulher” terá um valor próximo ao de “rei” e “rainha”.

2.3 Modelos clássicos

2.3.1 Bag-of-words

No *bag-of-words*, representa-se o texto (sentença ou documento) como um conjunto de palavras, desconsiderando sua ordem. Armazena-se também a frequência de ocorrência de cada termo (ou palavra), e utiliza-se essas informações como características nas tarefas. A seguinte fórmula descreve o método:

$$\mathbf{tf}(t, D) = \frac{\text{Número de ocorrências do termo t}}{\text{Número de termos no documento D}} \tag{2-2}$$

2.3.2 TF-IDF

O TF-IDF[45] é uma melhoria nas abordagens baseadas em frequência. Ela considera a frequência do termo, e a frequência inversa do documento (IDF). A primeira corresponde a equação 2-2 e a segunda é dada pela seguinte equação:

$$\mathbf{idf}(t, D) = log\frac{Número\ de\ documentos\ D}{Número\ de\ documentos\ contendo\ t} \tag{2-3}$$

A IDF visa medir o grau de relevância de uma palavra no *corpus* inteiro. Uma palavra que aparece em muitos documentos tende a ter baixa relevância, como exemplo, pode ser citado conectores como “mas”, “e”, “ou”.

O TF-IDF se trata de uma heurística composta de ambas as equações, resultando na seguinte multiplicação:

$$\mathbf{tf\text{-}idf}(t, d, D) = tf(t, d) \times idf(t, D) \tag{2-4}$$

2.4 Modelos de Redes Neurais

De acordo com [16], uma rede neural é um sistema computacional composto por uma série de unidades simples de processamento, que coletam e armazenam o conhecimento obtido a partir de um conjunto de dados de entrada. As redes neurais se constituem em uma das estruturas capazes de dar suporte ao Aprendizado de Máquina. Dentre as várias formas que as redes neurais podem assumir estão aquelas em que se utiliza de estruturas de aprendizagem profunda [17]. Na aprendizagem profunda, as redes neurais possuem muitas camadas ocultas, compondo uma arquitetura capaz de extrair características complexas dos dados [18].

As redes neurais mais comumente utilizadas são as *totalmente conectadas*, caracterizada por camadas que são, na prática, matrizes calculadas a partir das camadas predecessoras. Elas são compostas de camadas de entrada e de saída,

Tabela 2.1: Funções não lineares

Função	**Fórmula**
Softmax	$f(x) = \frac{exp(x_j)}{\sum_{k=1}^{K} exp(z_k)}$
Sigmoid	$f(x) = \frac{1}{1+exp(-x)}$
Tahn	$f(x) = \frac{exp(x)-exp(-x)}{exp(x)+exp(-x)}$
ReLU	$f(x) = max(0, x)$

com n camadas entre as duas, chamadas de camadas escondidas. Uma função não linear é geralmente utiliza na saída das camadas ou na saída da camada final, dependendo da aplicação. Algumas das mais comuns são mostradas na Tabela 2.1.

Nas redes neurais existem dois tipos de operações básicas: a inferência, que computa uma saída (que serve para regressão ou classificação), e a retro-propagação, que computa o erro da inferência em comparação com a saída esperada, e faz o ajuste dos pesos.

As redes neurais apresentam alguns problemas extensamente documentados. Dentre eles, a explosão e sumiço de gradientes são problemas notáveis.

A explosão de gradientes ocorre quando gradiente de erros grandes se acumulam e resultam em atualizações grandes aos pesos da rede neural durante o treinamento. Quando uma magnitude desses gradientes se acumulam, a rede se tornará instável, causando uma predição ruim e até inútil. Já o sumiço de gradientes ocorre quando os gradientes se movem exponencialmente para zero, a medida que os pesos são atualizados com a retro-propagação.

2.4.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) são redes neurais inspiradas no comportamento do córtex visual humano [18]. Dois dos seus tipos mais distintos de camadas são a camada convolucional, que realiza a operação de convolução na imagem, e a camada de *pooling*, que reduz a dimensionalidade da entrada ao tomar a média ou o máximo regional. Ela também faz uso de camadas não lineares como a relu, e camadas totalmente conectadas. A Figura 2.1 ilustra uma arquitetura de rede convolucional.

Na camada convolucional, cada neurônio está apenas conectado a um pequeno subconjunto de neurônios locais da camada anterior, que são uma matriz quadrada com altura e largura pré-definidos, também conhecida como "campo receptivo". Esses neurônios se estendem até a profundidade do volume de entrada, sendo extratores de características que trabalham somente para reconhecer padrões locais, reduzindo a complexidade de processamento que

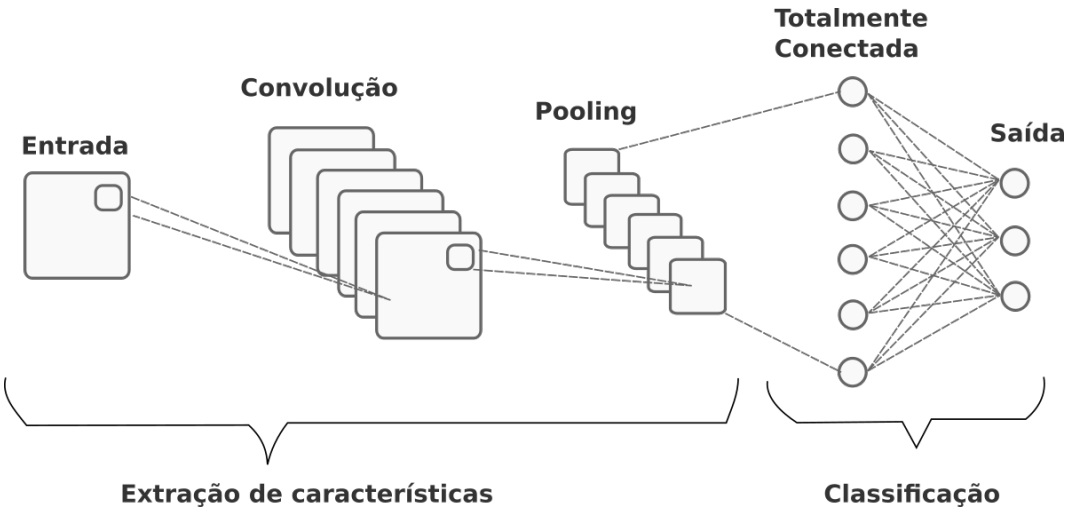


Figura 2.1: Rede Convolucional. Adaptado de [34]

uma rede totalmente conectada teria, por exemplo. O conjunto dos pesos desses neurônios é conhecido como *filtros* ou *kernels*. A saída da camada convolucional é conhecida como *features maps*.

A camada de *pooling* é uma forma de sub-amostragem não linear, que reduz a representação espacial da entrada. Um exemplo comum é o *max pooling*, que particiona a entrada em retângulos e tem como saída o máximo de cada um deles. Benefícios obtidos ao utilizar essa operação são a diminuição de parâmetros, quantidade de computação e menor possibilidade de *overfitting*

2.4.2 Recurrent Neural Network

As redes neurais recorrentes [2] [32] são uma arquitetura de redes neurais desenvolvida para lidar com dados e tarefas em sequências, como modelagem de linguagem e sumarização de texto[46]. Formalmente, uma RNN tem como entrada uma sequência de vetores $x_1, ..., x_n$ processados um de cada vez, onde para cada entrada x_n a rede atualiza seus parâmetros, e de maior importância, atualiza seu *estado escondido* utilizando a saída dos estados escondidos anteriores como entradas adicionais. O estado escondido constrói assim uma forma de memória, podendo lidar com informações parciais. A cada passo, uma RNN computa o estado escondido hn como:

$$h_n = RNN(\mathbf{h}_{n-1}, \mathbf{x}_n)$$

$$= f(\mathbf{W_{hh}}\mathbf{h}_{n-1} + \mathbf{W_{hx}}\mathbf{x}_n + \mathbf{b_h})$$

onde x_n é a entrada no passo n, h_{n-1} é o estado escondido anterior, f é uma função de ativação não-linear, e W, b são os parâmetros da RNN. Em

cada passo a RNN gera na sua saída um simbolo discreto y_n amostrado de uma distribuição de probabilidade por uma operação como o *softmax*:

$$y_n \sim softmax(\mathbf{W_y}\mathbf{h}_n + \mathbf{b_y})$$

onde a entrada(transformação linear de $\mathbf{h}_m$) é um vetor de importância acerca de cada classe da saída, e que é convertido em um vetor de probabilidades pelo *softmax*.

As RNNs possuem como maior vantagem a sua habilidade de captura dependências de longo termo nas sequências, graças a suas conexões recorrentes. No entanto, ela apresenta duas desvantagens já citadas anteriormente: explosão e sumiço de gradientes [33].

No caso da explosão de gradiente, técnicas como *gradient norm clipping*, que reescalam os gradientes, para que a norma deles não ultrapasse um valor determinado, são efetivas na prática. No caso do sumiço de gradientes, esse problema torna a rede recorrente inefetiva em aprender dependências de longo termo. Várias soluções foram propostas, dentre as mais populares são as de substituir a unidade recorrente por uma *long short-term memory* (LSTM) ou uma *Gated Recurrent Unit* (GRU). A ideia principal é utilizar unidades de portões que controlam a quantidade de memória que a RNN deve usar dos passos anteriores (*forget gates* $\mathbf{g}_n^{\mathbf{f}}$), o quanto receber do sinal de entrada (*input gates* $\mathbf{g}_n^{\mathbf{i}}$, e o quanto extrair de informação (*output gates* $\mathbf{g}_n^{\mathbf{o}}$) em cada passo. Todos as unidades de portões são computadas como transformações lineares da entrada atual x_n e estado escondido anterior $\mathbf{h}_{n-1}$ seguidos pela função de ativação do *sigmoid*:

$$\mathbf{g}_n^{\mathbf{i}} = \sigma(\mathbf{W_{ih}}\mathbf{h}_{n-1} + \mathbf{W_{ix}}\mathbf{x_n} + \mathbf{b_i})$$

$$\mathbf{g}_n^{\mathbf{f}} = \sigma(\mathbf{W_{fh}}\mathbf{h}_{n-1} + \mathbf{W_{fx}}\mathbf{x_n} + \mathbf{b_f})$$

$$\mathbf{g}_n^{\mathbf{o}} = \sigma(\mathbf{W_{oh}}\mathbf{h}_{n-1} + \mathbf{W_{ox}}\mathbf{x_n} + \mathbf{b_o})$$

A LSTM é definida, além das três unidades de portões, pelas fórmulas:

$$\mathbf{g}_n = \sigma(\mathbf{W_{gh}}\mathbf{h}_{n-1} + \mathbf{W_{gx}}\mathbf{x}_n + \mathbf{b_g})$$

$$\mathbf{c}_n = \mathbf{gf}_n \odot \mathbf{c}_{n-1} + \mathbf{g^i}_n\mathbf{g_n}$$

$$\mathbf{h}_n = \mathbf{g^o}_n \odot tanh(\mathbf{c}_n)$$

onde $\mathbf{c}_n$ é célula de estado que serve para armazenar informação temporal, e $\odot$ denota a multiplicação da matriz elemento por elemento.

A Rede Neural LSTM possui uma performance prática superior as RNNs,

sendo uma escolha popular atualmente para lidar com problemas sequencias como processamento de texto. Uma representação das células é mostrada na Figura 2.2.

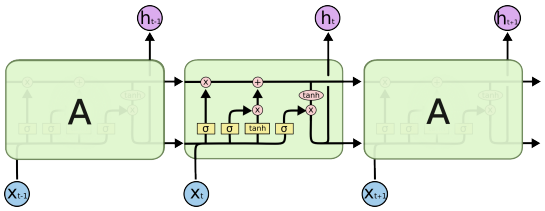


Figura 2.2: Célula LSTM. Fonte: [35]

2.4.3 Redes bidirecionais

As redes recorrentes RNN e LSTM geralmente processam os dados em um sentido (esquerda para direita). Redes neurais recorrentes bidirecionais são uma melhoria dessa forma básica, onde se tem duas redes lendo nas duas direções, permitindo que a saída consiga informação do passado e dos estados futuros simultaneamente. Apesar do custo adicional em performance, bons resultados foram obtidos com essa arquitetura, em especial o ELMO [29].

2.4.4 Encoder-decoder

Em um RNN Encoder-Decoder, temos o *encoder* que lê a sequência de entrada $x_1, ...x_n$, e codifica em um vetor de contexto $\mathbf{c}$. Dado a entrada de uma RNN, temos que:

$$\mathbf{c} = q(h_1, ...h_n)$$

onde q é uma função não-linear. O *Decoder* é geralmente treinado para predizer a próxima palavra y_t dado o contexto $\mathbf{c}$ e todas as palavras anteriores $y_1, ..y_{t-1}$. formalmente, o *decoder* é definido como:

$$p(\mathbf{y}) = \sum_{t=1}^{T} p(y_t|(y_1, ..., y_{n-1}), \mathbf{c})$$

onde $\mathbf{y} = (y_1..., y_n)$

Dessa forma, a arquitetura é capaz de lidar com tarefas que exigem transformações de sequência para sequência, como *machine translation*, por exemplo.

2.4.5 Attention

Nos modelos recorrentes, assume-se que, ao alimentar a entrada sequencialmente, ele será capaz de ter um estado escondido contendo a informação necessária dessa sequência para a tomada de decisão. Na prática, no entanto, em muitos casos essa condição é difícil de se garantir. O mecanismo de atenção (*Attention*) [5], que foi proposta para a tarefa de *machine translation*, relaxa essa condição, permitindo o acesso a informação do estado escondido em cada unidade. Ela passa a decidir quais estados escondidos se deve dar mais atenção. Em uma arquitetura *encoder-decoder* para a tarefa de MT, propõe-se o uso de um vetor dinâmico de contexto *cm* computado como a combinação linear ponderada do estados escondidos produzidos pelas RNNs, como:

$$c_m = \sum_{N}^{n=1} \alpha_{mn} \mathbf{h}_n \tag{2-5}$$

onde α_{mn} é o peso associado a cada representação $\mathbf{h}_m$. Os pesos são computados por:

$$\alpha_{mn} = \frac{exp(e_{mn})}{\sum_{n'=1}^{N} exp(e_{mn'})}$$

$$e_{mn} = \mathbf{v}^T tanh(\mathbf{W_{ah}}\mathbf{h}_n + \mathbf{W_{as}}\mathbf{s}_{m-1})$$

e pode ser pensado como a probabilidade de alinhamento entre um simbolo alvo na posição m e um simbolo fonte na posição n. e_{mn} é o *score* de alinhamento que diz quão bem as entradas na posição n e a saída na posição m casam. Ele é calculado com base nos estado escondido anterior s_{i-1} e na representação h_n da palavra fonte na posição n, sendo referida como atenção aditiva.

O modelo de alinhamento é parametrizado por uma rede totalmente conectada, treinada com os demais componentes da arquitetura. A Figura 2.3 ilustra o uso de *attention* em uma rede recorrente bidirecional. Sua utilização eventualmente se estendeu para além da tarefa de *machine translation*, como veremos nas próximas seções.

2.4.6 Scaled Dot-Product Attention

A *attention* pode também ser formalizada em termos de consulta (*query*), chave (*key*) e valor (*value*). Considera-se que $\mathbf{q}$ seja o vetor de *query*, e (k_i, v_i) sejam os pares *key-value*. Na atenção *Scaled Dot-Product* [19], representada na Figura 2.4 computa-se a atenção ao fazer o produto interno da *query* com as chaves, dividindo por $\sqrt{d_k}$, e aplicando-se a função *softmax*, como em:

$$Attention(Q, K, V) = softmax(QK^T\sqrt{d_k})V, \tag{2-6}$$

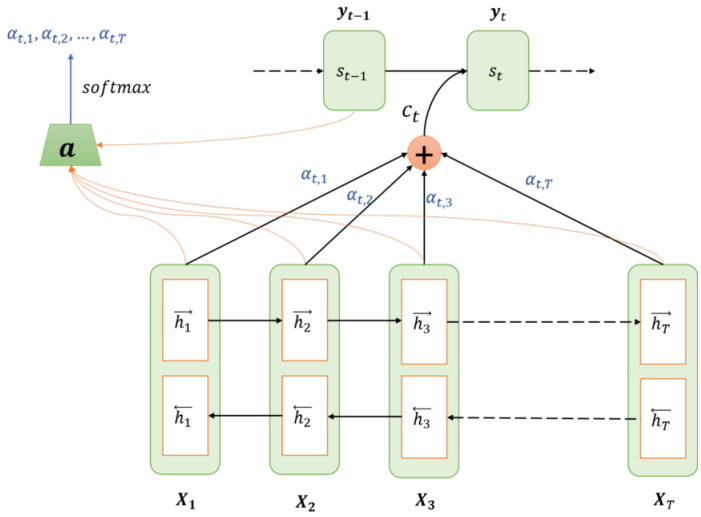


Figura 2.3: *Attention*. Fonte: [36]

onde assumindo que q e k são vetores de dimensão d_k com seus componentes sendo variáveis aleatórias de média 0 e variância 1, então o produto interno de $q \cdot k$ teria média 0 e variância d_k. Para que os valores tenham variância, dividimos por $\sqrt{d_k}$

A vantagem de usar atenção *dot-product* ao invés da atenção aditiva, é que a primeira é mais rápida e mais eficiente, já que pode ser implementada ao se utilizar operações otimizadas de multiplicação de matrizes.

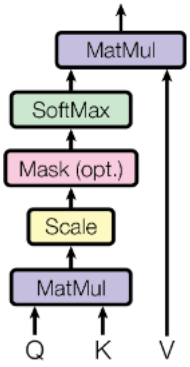


Figura 2.4: *scaled dot attention*. Fonte: [19]

2.4.7 Multi-Head Attention

Ao invés de utilizar uma única função de *attention*, o *Multi-Head Attention* [19] propõe projetar as entradas linearmente H vezes. Em cada uma dessas entradas, as atenções são calculadas, e seus resultados concatenados. Essa operação permite que o modelo possa obter informações conjuntas de

diferentes posições da sequência. Sua fórmula é dada por:

$$MULTIHEAD(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{W^O} Concatena(\mathbf{head}_1; ...; \mathbf{head}_H)$$

onde $head_h = Attention(W_h^Q Q, W_h^K K, W_h^V V)$ e W são todas matrizes com parâmetros. A atenção é presentada na Figura2.5

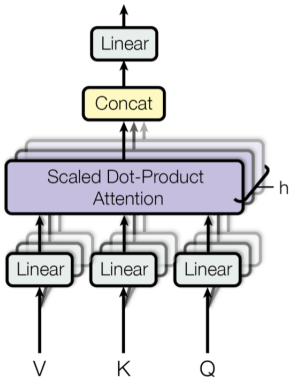


Figura 2.5: *Multi-Head Attention*. Fonte: [19]

2.4.8 Transformers

O *transformers* [19] é uma arquitetura composta primordialmente de módulos de *attention* puros, chamados *self-attention*, que funcionam sem o propósito de auxílio a outra camada como redes recorrentes. Além disso essa arquitetura faz uso de *multi-head self-attentions*, e é divida entre blocos de codificadores e decodificadores.

Alguma características adicionais:

- Nas camadas de atenção *encoder-decoder*, as *queries* vem da camada de decodificação anterior, e as *keys* e *values* vem da saída do codificador. Assim cada posição do decodificador poderá acessar todas as posições da entrada.
- Em uma camada *self-attention* do codificador todas as *queries*, *keys* e *values* vem da saída do codificador da camada anterior. Cada posição no codificador pode acessar cada posição do codificador da camada passada.
- Uma camada *self-attention* do decodificador também permite acesso a camada anterior como no codificador. Além disso, o fluxo de informação vindo da esquerda deve ser impedido, para se manter a propriedade auto-regressiva. Para isso, cria-se uma máscara na *scaled-dot attention* que substitui conexões ilegais por $-\infty$

Além disso, as camadas dos *encoders* e *decoders* possuem sub-camadas *fully-connected*, faz uso de *skip-connection* e camadas de normalização, como mostrado na Figura 2.6

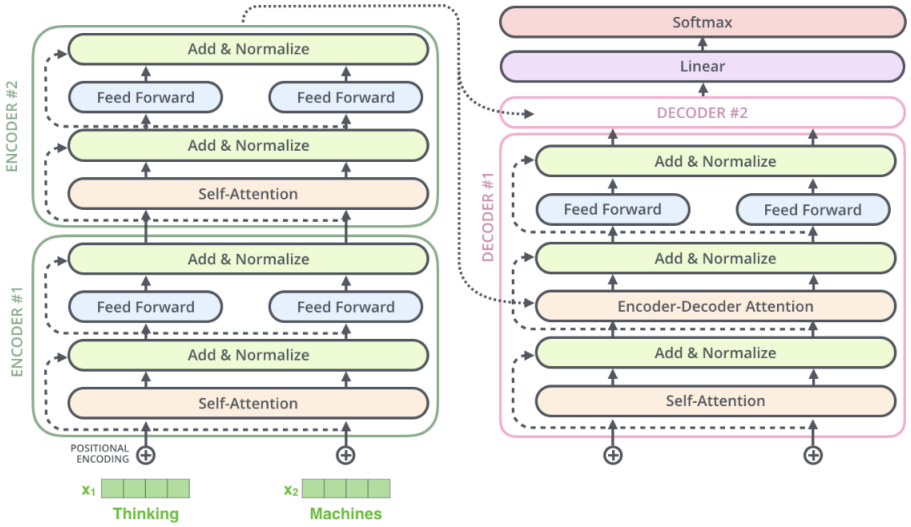


Figura 2.6: *Transformer*. Fonte: [37]

2.5 Modelos de Linguagem Estáticos

Nesta seção, são apresentados os modelos que aprendem um espaço vetorial de similaridades entre palavras, desconsiderando seu contexto, ou seja, são vetores estáticos.

2.5.1 CBOW e Skip-Gram

O CBOW e Skip-gram são duas estratégias de treinamento para aprendizado de vetores de palavras de palavras em modelos de aprendizado de máquina a partir de uma corpora. Eles exploram a hipótese que palavras que pertencem ao mesmo documento, ou a uma mesma janela de palavras, possuem uma maior similaridade entre si do que as de documentos diferentes ou fora da janela. Ambas as estratégias tentam aprender a similaridade das palavras explorando essa hipótese. Elas são retratadas em 2.7.

No Skip-gram, o vetor de palavra da palavra de entrada é utilizada para predizer o contexto, isso é, uma janela de palavras vizinhas. Dado uma sequencia de palavras $w_1, w_2, w_3, ..., w_T$, o objetivo do modelo Skip-gram é maximizar a probabilidade logarítmica média

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} logp(w_{t+j}|w_t) \tag{2-7}$$

onde c é o tamanho do contexto de treinamento(em função da palavra central w_t). O termo $p(w_{t+j}|w_t)$ pode ser definido pela função softmax:

$$p(w_O|w_I) = \frac{exp({v'_{w_O}}^{\top} v_{w_I})}{\sum_{w=1}^{W} exp({v'_w}^{\top} v_{w_I})} \tag{2-8}$$

onde v_w e v'_w são as representações vetoriais de entrada e saída respectivamente de w, e W é tamanho do vocabulário. Como o custo do logaritmo dessa função vai ser proporcional ao tamanho do vocabulário, essa abordagem é inviável. Uma aproximação chamada Negative sampling[ref] é comumente utilizada. Nela, somente uma parcela dos pesos é atualizada ao se modificar a tarefa de aprendizado, definida como:

$$log\sigma({v'_{w_O}}^{\top} v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)}[log\sigma(-{v'_{w_i}}^{\top} v_{w_I})] \tag{2-9}$$

Onde o primeiro termo maximiza a probabilidade de ocorrência das palavras corretas na janela de contexto, permitindo que co-ocorram. O segundo termo itera sobre algumas palavras que não ocorrem na janela, e minimizam suas probabilidades de co-ocorrência. Desse modo o modela evita ter que atualizar todos os neurônios representantes de palavras que não tem influência com a janela sendo treinada.

Já no CBOW, os vetores de palavras das palavras vizinhas são utilizadas para prever a palavra central. Ele é consideravelmente mais rápido que o Skip-gram, com uma reportada melhor acurácia em palavras frequentes.

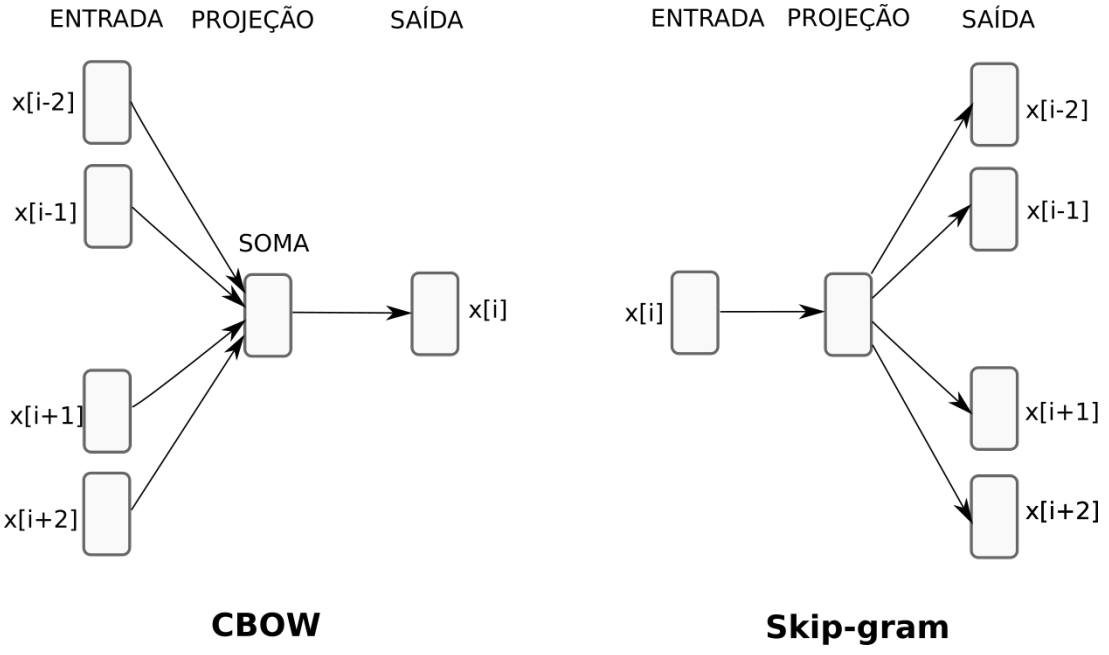


Figura 2.7: CBOW e Skip-gram. Adaptado de [38]

2.5.2 Word2Vec

O *word2vec* é uma abordagem de treinamento de redes neurais para geração de vetores de palavras de palavras. Ele foi proposto por [28], e implementa as duas estratégias de treinamento de embeddings apresentadas: CBOW e Skip-gram. É geralmente uma rede com duas camadas escondidas totalmente conectadas, sendo que a primeira tem a sua dimensão definida pelo tamanho do vocabulário(número de palavras relevantes da corpora) e a segunda com um tamanho a ser estabelecida, e com uma função não-linear na saída.

2.5.3 FastText

Uma limitação do *word2vec* é ser incapaz de gerar vetores de palavras para palavras fora do vocabulário em que o modelo foi treinado. *Fasttext* ameniza esse problema ao aprender informações da estrutura interna das palavras. O modelo aprende vetores *n-grams* de caracteres, e dependendo do n, divide as palavras em termos como: "com-endo", "faz-endo". O vetor de palavra de uma palavra vai ser dado pela soma das representações vetoriais dos *n-grams*, permitindo uma representação compartilhada entre palavras.

2.5.4 Glove

Glove[77] é um método estatístico que incorpora a informação local com janela de palavras, mas que também faz uso de informações globais ao construir uma matriz M de co-ocorrência das palavras. Então, em cada elemento M_{ij} é representada a probabilidade da palavra i ser próxima de j. Os vetores da matriz são gerados aleatoriamente e treinados com a equação

$$P(v_i, v'_j) = log(M_{ij}) = v_i v'_j + b_i + b_j \tag{2-10}$$

onde b_i e b_j são bias e v_i e v'_j são os vetores de palavras. Intuitivamente, o produto interno dos dois vetores é otimizado para se correlacionar com a probabilidade de sua co-ocorrência. Essa formulação ainda é melhorada no trabalho original com mais dois fatores: primeiro, o $log(M_{ij})$ diverge quando não há co-ocorrências entre i e j, portanto a conversão log(x) para log(x+1) é feita. Segundo, co-ocorrências muito frequentes não são muito relevantes, sendo proposto uma função $f(M_{ij})$ de peso para compensar esse desvantagem. Essa função é definida como:

$$f(x) = \begin{cases} (x/x_{max})^{\alpha}, & \text{se } x < x_{max} \\ 1, & \text{senão} \end{cases} \tag{2-11}$$

Por fim, converte-se a equação principal para um problema de otimização dos mínimos quadrados, adicionando as melhorias e tendo assim a equação final do modelo *glove*:

$$J = \sum_{i,k=1}^{V} f(M_{ik})(v_i v_j' + b_i + b_j - log(1 + M_{ij}))^2 \tag{2-12}$$

2.6 Modelos de Linguagem Contextuais

Os modelos apresentados anteriormente desconsideram o contexto ao computar os vetores de palavras de palavras. Essa limitação impacta no entendimento de frases que contem palavras polissêmicas. Por exemplo, a palavra "apple" pode ter dois significados diferentes de acordo com o contexto: como fruta ou como a empresa. Os modelos dependentes de contexto foram elaborados em suas arquiteturas e estratégias de treinamento para considerar tais casos.

Um exemplo de modelo de linguagem dependente de contexto é o ELMO, que faz uso de redes Bi-LSTM, que são compostas de uma LSTM normal e outra invertida, e aprende *features* contextuais sensíveis. A representação de cada *token* será a concatenação das representações esquerda-pra-direita e direita-pra-esquerda, tendo assim a contextualização necessária.

Outros modelos que consideram o contexto tem sido desenvolvidos, e dentre eles um de notável destaque é o modelo *transformer* profundo bidirecional, conhecido como BERT, o qual será abordado na próxima seção.

2.6.1 Transformer-BERT

Modelos *transformers* tem tido bastante sucesso em diversas tarefas de NLP. No entanto, antecedente ao BERT, os modelos de linguagem eram unidirecionais, ou seja, liam os dados de forma sequencial em uma direção somente, geralmente da esquerda para a direita, fazendo com que nas camadas de self-attention só pudessem atender a tokens anteriores. Um exemplo de transformer unidirecional é o GPT[52]. O BERT trata essa limitação com o uso da função objetivo de pré-treinamento chamada *masked language model*.

A função de objetivo *masked language model* funciona da seguinte forma: Ela esconde uma parcela dos *tokens* de entrada (15% geralmente), e define que a treinamento do modelo seja de prever as *tokens* escondidas. Além dessa, o

modelo também é treinado na tarefa de predição da próxima sentença, onde dado duas sentenças A e B, 50% das vezes B será a proxima sentença, e 50% será outra sentença aleatória, e a tarefa é predizer se B é a próxima sentença. Essas estratégias de treinamento foram utilizadas com sucesso em outros modelos *transformers* como RoBERTa [21]. Uma arquitetura básica do BERT é ilustrada em alto nível na Figura 2.8.

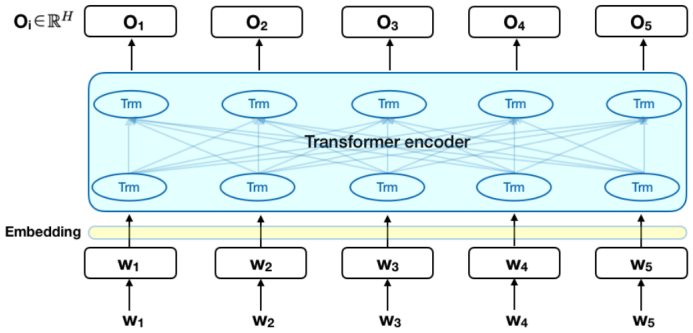


Figura 2.8: Modelo BERT(link)

É necessário frisar no entanto que esses modelos exigem muito recursos computacionais, tendo que ser treinados em hardware especializado (e.g. GPUs ou TPUs, por exemplo). Uma vantagem deles é que uma vez treinados, podem ser *finetuned* em tarefas consideradas *downstream*, como classificação de texto, NER etc.

2.7 Modelagem de Tópicos - LDA

Tópicos em aprendizado de máquina são conjuntos de palavras com um ou mais temas similares. Um modelo de Tópico tem então o propósito de descobrir tópicos abstratos em um corpus de documentos. Como exemplo, na Figura 2.9 é retratado a estrutura de um modelo de tópico. Assume-se que um documento possui diferentes tópicos, e que cada tópico é uma distribuição de probabilidade sobre palavras

O *Latent Dirichlet Allocation* (LDA)[56] é uma família de modelos geradores estatísticos muito utilizados para encontrar variáveis latentes que explicam documentos ou dados. Eles são considerados modelos misturados[57], o que significa que os documentos que processam podem pertencer a múltiplos tópicos. Além disso cada tópico é uma mistura de palavras, onde as palavras podem ser compartilhadas entre tópicos. Assim, um único documento pode pertence a múltiplos tópicos, cada qual com uma probabilidade associada. Em sua forma simples, o LDA é um modelo *bag-of-words* onde cada vetor é uma probabilidades dos termos sobre um vocabulário global.

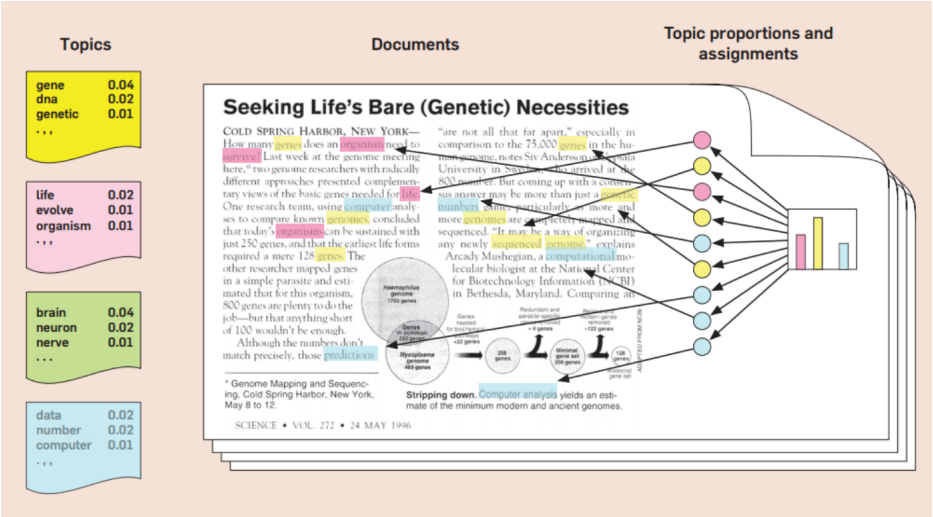


Figura 2.9: Tópicos em um documento. De [69]

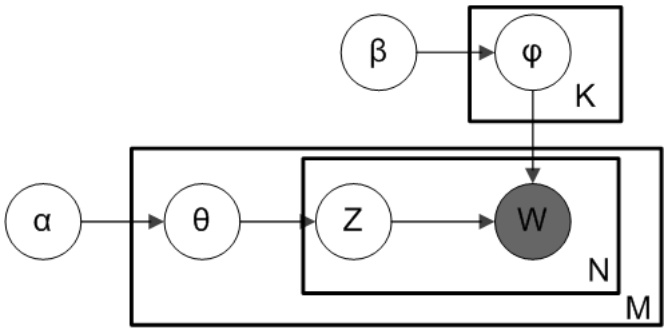


Figura 2.10: LDA

O funcionamento do LDA pode ser descrito pela notação de placas na figura 2.10. Resumidamente:

- Dada um número de documentos M, um número N de palavras por documento, e um conjunto de palavras W, procura-se descobrir as variáveis latentes θ e Z, que representam a distribuição de documentos e tópicos respectivamente.
- O processo gerador assumido pelo modelo é o seguinte: Ao amostrar da distribuição de tópico por documento θ, é obtida uma atribuição de tópico z_n para a n-ésima palavra. Ao amostrar do tópico z_n, obtém-se uma palavra w_n. Dado que a única variável observável é W, o objetivo então se torna aprender as probabilidades *a posteriori* para as demais variáveis latentes.
- No LDA, um priori esparso *Dirichlet* é proposto para modelar a distribuição de tópicos e palavras. φ é a distribuição multinomial de K tópicos com parâmetro β para amostragem de palavras. θ é a distribuição multi-

nomial de D documentos com parâmetro α para amostragem de tópicos .

– Vários métodos são proposto para estimar os parâmetros das distribuições, dentre eles *Gibbs Sampling*[58] é bastante utilizada.

2.8 Redução de Dimensionalidade - UMAP

O *Uniform Manifold Approximation and Projection for Dimension Reduction* (UMAP)[60] é uma técnica de redução de dimensionalidade popular para visualização de dados. Mais sucintamente, ele encontra uma representação de baixa dimensionalidade dos dados com propriedade topológicas similares a sua versão de alta dimensionalidade. Ele constrói um grafo de relações locais entre os dados no espaço de alta dimensionalidade, e então utiliza otimização para encontrar vetores de palavras no espaço de baixa dimensionalidade que preserve a estrutura desse grafo. O caso mais usado de espaço de baixa dimensionalidade é o de duas dimensões, que permite a visualização e interpretação dos dados.

Devido ao alto custo computacional do UMAP, ele se torna inviável em *datasets* de grande volume, como é o caso das teses/dissertações. O UMAP paramétrico [41] torna o algoritmo mais eficiente ao fazer usos de redes neurais que substituem as etapas mais pesadas. Como se vê na Figura 2.11, as etapas de construção do grafo e geração de vetores de palavras que preservam a estrutura do grafo são substituídas por redes neurais.

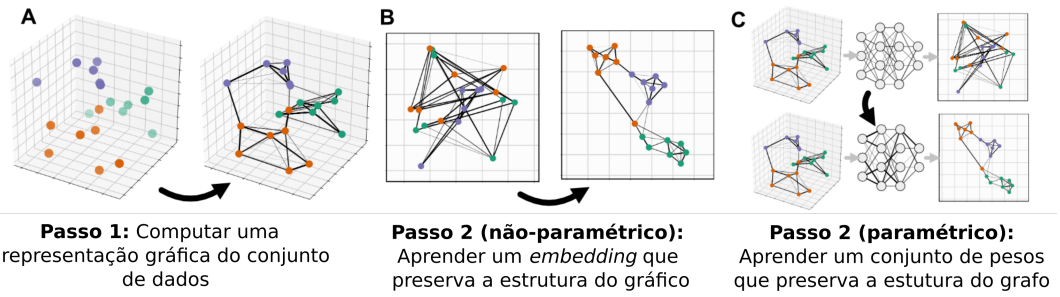


Figura 2.11: Processo do UMAP. Traduzido de [41]

2.9 Tarefas de PLN

A área de processamento de linguagem natural possui uma vasta gama de tarefas. Aqui são relatadas algumas das mais relevantes nesse trabalho.

2.9.1 Classificação de Texto

A classificação de texto é uma das tarefas primordiais do PLN, sendo a classificação de documentos uma subcategoria de bastante interesse. A tarefa em sua forma tradicional pode ser dividida nas etapas:

1. Tem-se documentos textuais na forma $D = X_1, X_2, ..., X_n$, onde X_i se refere a uma amostra (documentos, sequência de texto).

2. Os documentos são pré-processados, retirando-se símbolos e caracteres que não sejam de interesse a tarefa, bem como normalizando o texto se necessário, dentre outras abordagens.

3. Realiza-se a extração de características relevantes, como o *Term Frequency-Inverse Document Frequency* (TF-IDF) por exemplo.

4. Se necessário, utiliza-se algoritmos de redução de dimensionalidade para uma melhor performance na etapa de classificação.

5. Realiza-se o treinamento de algum dos métodos supervisionados de aprendizado de máquina.

6. As métricas da classificação são computadas

A categorização de documentos também pode ser subdividida em:

- Nível de Documento: Utiliza-se todo o documento para categorização
- Nível de Parágrafo: Utiliza-se um parágrafo para categorização
- Nível de Sentença: Utiliza-se uma sentença (parte de um parágrafo) para categorização
- Nível de Sub-Sentença: Utiliza-se sub-expressões de uma sentença para categorização.

2.9.2 Geração de Modelos de Linguagem

A tarefa de modelagem de linguagem consiste em se estimar a probabilidade conjunta de uma sequência de *tokens* $P(w_1, ...w_n)$ caracterizado como:

$$P(\mathbf{w}) = \prod_{n=1}^{N} P(w_n|w_1...w_{n-1}) = \prod_{n=1}^{N} P(w_n|\mathbf{w}_{<n}) \tag{2-13}$$

Na equação, temos a predição do próximo *token* dado o histórico de *tokens* previamente vistos, o que define um *Forward Languagem Model*. Já um *Backward Language Model* observa as *tokens* na ordem inversa, o que equivale a predizer as *tokens* dado o contexto futuro $P(w_n|\mathbf{w}_{>n})$.

2.9.3 Named Entity-Recognition

Com o objetivo de identificar nomes de pessoas, organização e localizações geográficas, durante a *Message Understanding Conference* [4] foi definido o termo *Entidade Nomeada* (ou NE, do inglês *Named Entity*). Uma entidade nomeada é uma forma de categorizar um elemento como pertencente a um grupo de elementos com atributos similares aos dele. Pode-se considerar como entidades a data, local, área de estudo, ou, até mesmo, entidades customizadas. Assim, a tarefa reconhecimento de entidade nomeada (ou NER, do inglês *Named Entity Recognition*) tem como objetivo identificar os elementos de interesse do texto e marcá-los de acordo com as categorias que correspondem às entidades nomeadas.

2.9.4 Question Answer

Sistemas pergunta-resposta (*Question Answering* – QA) representam uma subárea de processamento de linguagem natural relacionada à recuperação de informação. Eles têm como objetivo responder a uma pergunta com a recuperação de informações relevantes extraídas de um conjunto de dados existente.

Em [22] são definidos dois tipos de sistemas *question-answering* de acordo o com tipo de dado: *Knowledge Base Question Answering* (KB-QA), que utiliza bases de conhecimento como a DBPedia [23], YAGO [25] e Wikidata [24]; e o *Text-QA*, que extrai a resposta a partir de uma coleção de documentos. Além disso, o QA ainda pode ser classificado como turno simples, onde o usuário envia tudo o que deseja em uma única consulta, ou multi-turno (*Conversational QA*), que possibilita ao usuário decompor a sua pergunta em vários turnos.

3
Trabalhos Relacionados

Este capítulo contém o levantamento de alguns trabalhos na literatura que são relacionados ao tema proposto para esta dissertação. Objetivando selecionar aqueles fortemente relacionados com este trabalho, realizou-se uma seleção com base nas seguintes questões:

- Quais são os vetores de palavras existentes na língua portuguesa e seus domínios?
- Quais os esforços para construção de vetores de palavras ou *corpus* para o domínio acadêmico em português?
- Quais são os vetores de palavras e suas aplicações para o domínio acadêmico internacional existentes?

3.1
Processo de busca

Para a seleção dos artigos foi utilizado o motor de pesquisa *Google Scholar*,[1] que reúne um grande volume de produções científicas provindas de diferentes instituições, conferências e periódicos. Deste modo, o uso desses motores é importante no processo de busca de trabalhos da literatura que embasam esta dissertação. Foram considerados somente trabalhos divulgados a partir do ano de 2000, incluindo teses e dissertações.

Com o propósito de coleta dessas publicações, definiu-se um conjunto de termos considerados relevantes, filtrados de forma a selecionar apenas trabalhos publicados a partir de 2000. Foram construídas várias *strings* de busca, que foram ajustadas conforme a sintaxe do motor de busca utilizado. Para a primeira pergunta, para a qual se desejava os resultados que contivessem pelo menos uma das palavras, a esquerda da conjunção **E**, ocorrendo em conjunto com algum dos termos à direita, definiu-se a seguinte *query*:

- (("vetores de sentido" **OU** "vetores de palavras" **OU** "representação vetorial") **E** ("*portuguese*" **OU** "português"))

[1] `https://scholar.google.com.br/`

Para responder a segunda pergunta, formou-se uma *query* similar a anterior, só que priorizando a presença de termos relacionados ao domínio acadêmico, resultando na *query* a seguir.

– ((“vetores de sentido” **OU** “vetores de palavras” **OU** “representação vetorial”) **E** (“*portuguese*” **OU** “português”) **E** (“*acadêmico*” **OU** “científico”))

Para a ultima pergunta, construiu-se a *query* com termos em inglês

– (“*academic*” **OU** “*articles*” **OU** “*papers*” **OU** “*scientific*”) **E** (“embedding” **OU** “vetores de palavras” **OU** “*language model*”)

Foram realizadas consultas no *Google Scholar* de maneira semi-automatizada, considerando que os termos da busca poderiam ocorrer tanto no título quanto no resumo. Limitou-se os resultados até a décima página de cada consulta, ou seja, cem primeiros trabalhos (duzentos no total), visto que quanto mais distante da primeira página menor a relevância do conteúdo com a busca. Consequentemente é possível obter trabalhos fracamente relacionados à pesquisa. No intuito de lidar com esse e outros problemas, os resultados que se enquadravam nos seguintes critérios foram removidos:

1. Trabalhos que estejam escritos em outros idiomas que não sejam português ou inglês.
2. Trabalhos com assunto não relacionado às nossas perguntas.

Em seguida, foram analisadas as introduções e conclusões, filtrando de acordo com os critérios de exclusão definidos anteriormente. Posteriormente, manualmente foram recuperados e incluídos na análise os documentos que citam esses trabalhos, bem como os citados por eles, que fossem considerados mais relevantes.

3.2 Resultados

Após coletar os trabalhos e filtrá-los, os trabalhos que permaneceram no conjunto passaram a ser analisados mais profundamente, para se efetuar a escolha daqueles considerados mais relevantes.

3.2.1 Vetores de palavras na Língua Portuguesa

Da pesquisa feita sobre os trabalhos relacionados aos vetores de palavras na língua portuguesa, foram selecionados os considerados mais relevantes em termos de quantidade e qualidade.

Em [11] foi reunido um corpus de 1.7 bilhões de *tokens* em português brasileiro e europeu, a partir de artigos de notícias, fazendo um dos maiores corpus de dados existentes nessa linguagem, e treinaram um modelo *word2vec* usando as estratégias CBOW e Skip-gram. Subsequentemente, [14] adiciona 500 milhões de *tokens* coletados por 1 ano de sites de notícias, ao corpus anterior, obtendo 2 bilhões de *tokens*. Em [9] os autores fizeram *scrapping* de 60 milhões de páginas da web, e depois de filtragem e pré-processamento obtiveram 2.7 bilhões de *tokens*, nomeado o dataset de BrWaC. Posteriormente, [53] treinaram o modelo *transfomer* BERTimbaum no BrWaC. [8] coletaram 7.4 milhões de páginas de blogs em português brasileiro, obtendo um corpus com 2.1 bilhão de *tokens*.

Mais recentemente, [10] reuniu um grande corpus *multilingual* chamado OSCAR (*Open Super-large Crawled ALMAnaCH coRpus*), retirado do *Common Craw*, que é um depósito de páginas e documentos da *web*. O conteúdo disponibilizado em português contém cerca de 10 bilhões de *tokens*.

3.2.2 Vetores de palavras e *Corpus* no Domínio Acadêmico em Português

Da pesquisa feita sobre os vetores de palavras e *corpus* construídos para o domínio acadêmico na língua portuguesa, poucos resultados foram obtidos. Destacam-se aqui os encontrados.

O trabalho de [12], reuniu 47 introduções de dissertações de Mestrado e 3 introduções de qualificações de Mestrado, no domínio da Ciência da Computação, contendo no total 53.000 *tokens* e 1.350 sentenças. Com base nele, os autores propõem o uso subsequente de um analisador discursivo automático para o português.

[7] reuniu coleções de edições da revista científica FAPESP, obtendo cerca de 3,7 mil artigos, e um total de 500 mil *tokens*.

De domínios científicos mais especializados, destacamos o [6] que reúne um corpus em literatura biomédica e médica, com um total de 83 mil documentos e 18 milhões de *tokens*. [13] reuniu um corpus no domínio de óleo e gás, a partir de teses, dissertações, relatórios técnicos e glossários, obtendo cerca de 5 milhões de *tokens*. [73] construiu o corpus mais similar ao nosso, reunindo *abstracts* de teses e dissertações, acumulando um total de 34 milhões de *tokens*.

3.2.3 Vetores de palavras e suas Aplicações no Domínio Acadêmico Internacional

Por último, foi realizada uma busca pelos trabalhos internacionais que constroem vetores de palavras para o domínio acadêmico e suas aplicações como forma de guiar e contextualizar este trabalho. Os trabalhos encontrados foram majoritariamente na língua inglesa. Foi realizada uma filtragem pelos mais recentes e pelos considerados mais relevantes.

Em [31], os autores retreinam o modelo BERT, pré-treinado na Wikipédia e *BookCorpus*, em um corpus de artigos do *PubMed* com 18 bilhões de *tokens*, nomeando o modelo de BioBERT. Eles avaliam o modelo em tarefas de mineração de texto biomédica como NER biomédico, RE (Extração de relação) biomédico e QA biomédico, estabelecendo resultados estado da arte.

Em [30], os autores utilizam um corpus de 1.14 milhões de pré-prints, com 18% sendo da área de ciência da computação e 82% do campo da biomedicina. Foram contabilizados cerca de 3.17 bilhões de *tokens*. Os autores treinaram um modelo BERT inicializado com pesos aleatórios, o qual nomearam de SciBERT. O modelo foi avaliado em tarefas de classificação de texto e segmentação de texto (NER), e obteve melhores resultados que o BioBERT em algumas tarefas biomédicas.

Recentemente em 2020, com a crise de *Covid-19*, um esforço por parte do *Kaggle* e varias organizações estabeleceu um corpus de *pre-prints* científicos sobre o tema, descrito em [15]. Diversas tarefas foram elaboradas, geralmente voltadas com objetivo geral de recuperação de informação. O grande volume de pesquisa sendo gerada requer mecanismos que ajudam o entendimento e que realizem filtragem e buscas. Algumas tarefas conhecidas são *question answer*, *abstract* ou *text summarization*, dentre outras.

Os resultados comparados estão na Tabela 3.1. Como se pode observar, enquanto há uma disponibilidade significante de corpus em português de fontes mais generalistas, existe uma lacuna na disponibilidade de corpus acadêmicos para o português, como em comparação, por exemplo, ao inglês.

Tabela 3.1: Resumo dos trabalhos da literatura

Estudo	**Fonte**	**Idioma**	***Tokens***	**Domínio**
[11]	Notícias	português brasileiro e europeu	1.7 bilhões	geral
[14]	Notícias	português brasileiro e europeu	2.2 bilhões	geral
[9]	Paginas web	português brasileiro	2.7 bilhões	geral
[8]	Blogs	português brasileiro	2.1 bilhões	geral
[10]	Paginas web	português	10 bilhões	geral
[12]	Dissertações	português brasileiro	53 mil	acadêmico
[7]	Revista científica	português brasileiro	500 mil	acadêmico
[6]	Literatura médica	português brasileiro	18 milhões	acadêmico
[73]	Teses e Dissertações	português brasileiro	34 milhões	acadêmico
[13]	Documentos sobre óleo e gás	português brasileiro	5 milhões	acadêmico
[31]	Artigos do PubMed	inglês	18 bilhões	acadêmico
[30]	Semantic Scholar	inglês	3.17 bilhões	acadêmico

4
Construção do Corpus Acadêmico

A proposta principal deste trabalho é a construção de um *corpus* com conteúdo acadêmico/científico e o treinamento de modelos de linguagem de forma a se obter vetores de palavras especializados para esse domínio. Este capítulo encontra-se dividido em duas partes: a primeira abordando o processo de coleta e estatísticas do *corpus*, e a segunda tratando das técnicas propostas para os modelos de linguagem.

4.1
Construção do Corpus

A construção do corpus é definida pelas etapas representadas na Figura 4.1. A coleta dos arquivos em *CSV* e *JSON*, que possuem os links para a plataforma Sucupira, que por sua veze detém os arquivos de teses e dissertações. Após coletá-los, eles são alimentados ao GROBID, que extrai o texto dos PDFs em arquivos XMLs. Por fim, é realizado o *parsing* dos XMLs para extrair o texto relevante, dos quais são realizados limpeza e filtragem. As seções a seguir detalham todo esse processo.

4.1.1
Coleta

Para coleta foram utilizadas as tabelas em CSV do Catálogo de Teses e Dissertações que se encontram disponíveis para *download* no portal de dados

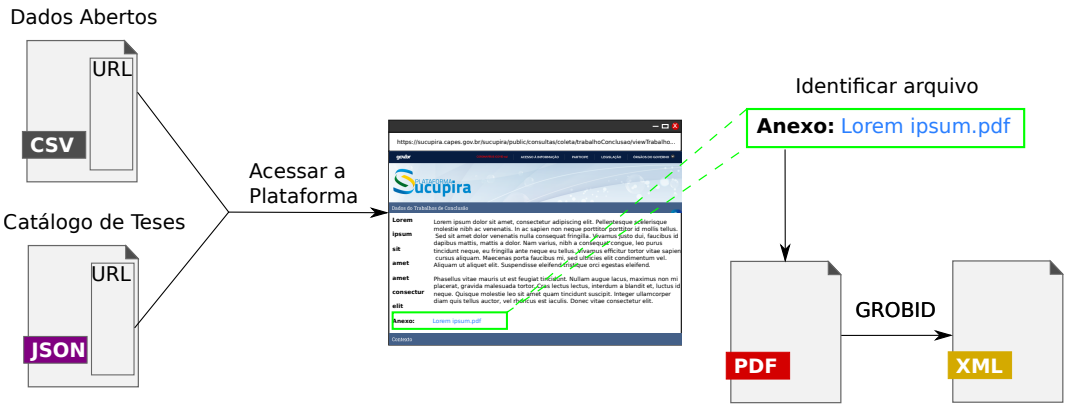


Figura 4.1: Processo de coleta das teses e dissertações

abertos da CAPES[1]. Nelas, estão armazenadas informações de trabalhos entre os anos de 1987 e 2019. Cada linha da tabela representa uma tese ou dissertação defendida junto a um programa de pós-graduação do Brasil. Dentre essas informações há o nome dos autores, área de conhecimento, resumo, link do texto completo.

Com base nas tabelas obtidas do portal de dados abertos, foi extraída a coluna referente aos links onde estão armazenados esses trabalhos. Grande parte desses links são da Plataforma Sucupira, e os demais são informações inválidas ou endereços para repositórios das universidades que representam a minoria dos links do conjunto. Com base nisso, foi selecionado apenas os registros que estão cadastrados na Plataforma.

O *scrapping* foi realizado por meio de *scripts* em *python* na página web do site sucupira a que os links levavam. Muitos dos links da plataforma Sucupira não possuem o PDF disponível no entanto. Foram baixados em torno de 350 mil PDFs a partir dos links contido nos CSVs do Catálogo de Teses e Dissertações dos Dados abertos.

Existe ainda outra fonte de dados com teses/dissertações mais recentes que 2019: o site Catálogo de Teses e Dissertações [2] que mantém informações de links atualizados de Teses/Dissertações na plataforma Sucupira. Os links mais recentes (2020+) não estão presentes nos CSVs. Foi feito o *scrapping* do site, e armazenado os links para Plataforma Sucupira em arquivos JSON. Assim, foi possível baixar cerca de 40 mil teses/dissertações. Esse número expressivo se deve ao fato que os links com PDFs mais recentes tem menor probabilidade de estarem ausentes.

Foi contabilizado um número final de teses/dissertações de 392mil. O tamanho do *corpus* foi calculado em aproximadamente 2 terabytes.

4.1.2 Grobid

O PDF[67] é um formato de arquivo planejado para ser independente de sistema operacional, hardware e aplicativos de softwares, sendo altamente utilizado para compartilhamento de documentos. Ele é baseado na linguagem *PostScript*[68], e cada documento PDF é composta na verdade de múltiplos arquivos como imagens,texto e fontes. Do ponto de vista do *machine learning*, PDFs são considerados como dados não estruturados, ou seja, não estão em um formato de *dataset* adequado para o aprendizados dos modelos. Para os modelos de linguagem, parte do conteúdo das teses/dissertações, como

[1] `https://dadosabertos.capes.gov.br/`
[2] `https://catalogodeteses.capes.gov.br/catalogo-teses/`

imagens e tabelas, não acrescentam informações significativas no aprendizado. É necessário então a extração do conteúdo textual por meio de ferramentas de extração.

Seguindo trabalhos que exploram as principais técnicas e ferramentas para extração de conteúdo, tem-se o consenso que a extração PDFs gerais não tem boa performance em documentos científicos, que precisam de técnicas e ferramentas especializadas que considera o *layout* desses tipos de documentos [49][50]. Dentre as ferramentas pesquisadas, a que mais se aproxima do problema em questão, que é a extração de conteúdo textual de teses/dissertações, é o Grobid.

O Grobid [26], é uma biblioteca bastante usada para extração de texto científico. Ela foi recentemente utilizada para extrair texto dos *pre-prints* de artigos sobre COVID-19 [15]. Ele é composto de modelos em cascata para rotulação de sequências de *tokens*, similar a tarefa de NER, e faz uso de *Deep Learning* e CRF [51] para tal. É definida uma *pipeline* de tarefas, representada na Figura 4.2 para cada modelo com objetivo de identificar diferentes seções do PDF.

O modelo *fulltext* tenta reconhecer e estruturar itens que aparecem no corpo do documento, dentre elas:

- Parágrafos
- Títulos da seção
- Figuras
- Tabelas
- Fórmulas
- Lista de itens
- Marcadores: chamada para figuras (e.g. “veja Figuras 1”) e referências bibliográficas

Existem também os modelos de Data, Figura e Tabela, que são chamados pelos demais modelos para detectar e segmentar todo tipo de data, imagens e tabelas no documento inteiro. O fato de serem reusáveis e especializados em suas respectivas tarefas melhora a performance do sistema como um todo.

Enquanto também foram avaliadas outras ferramentas de extração de PDFs gerais e científicos, nos experimentos realizados, o Grobid foi a biblioteca com os melhores resultados em diminuir erros gramaticais e em identificar texto de tabelas, mantendo a estrutura da tese/dissertação.

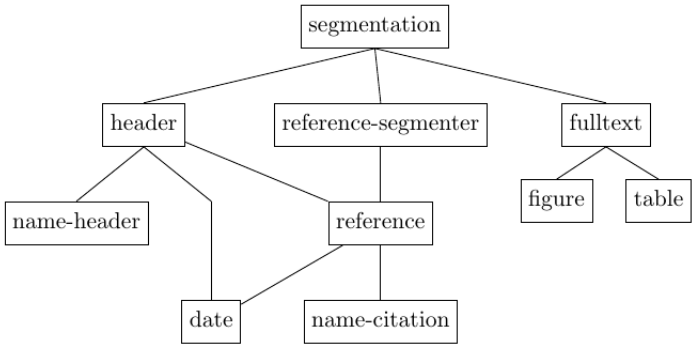


Figura 4.2: Fluxo e atributos da ferramenta Grobid. Link

4.1.3 Filtragem e Limpeza dos dados

Os arquivos resultantes tem o formato XML, estruturados nas *tags* previamente citadas, como cabeçalhos, títulos e parágrafos. Dado que se obteve essas anotações, a tarefa então se torna em quais anotações extrair e como lidar com os erros nas tags.

Foi desenvolvido um *script* de *parsing* em *python* com regras que buscam fazer essa duas funções: selecionar as *tags* de interesse, e filtrar as que estão erradas. Para a Seleção:

- Primeiramente, excluir todos os *XMLs* corrompidos(que possuem 0 bytes)
- Seleciona-se as *tags* de parágrafo somente

Para a filtragem foram criadas regras com base em observações de tipos de erros comuns que aconteciam. Elas são:

- Remoção do excesso de espaços em branco contidos em um parágrafo. Ex: “Ele comeu uma batata” se torna “Ele comeu uma batata”
- Remover parágrafos que possuem uma proporção maior que 30% de caracteres isolados. Ex “E l e co m e u u m a b a t a t a”
- Remover parágrafos com uma quantidade n de símbolos diferentes de letras e números. Ex: “E#& Co##u um! b@t@t@”
- Retirar parágrafos com uma proporção de números maior que 30%. Esse é o caso de teses/dissertações com fórmulas matemáticas extensas. Infelizmente muitos símbolos foram trocados pelo Grobid, as tornando inutilizáveis.

Enquanto as teses/dissertações com fórmulas matemáticas extensas sofreram os maiores cortes, o resultado final do corpus foi considerado satisfatório para o aprendizado de modelos de linguagem, com uma maioria legível e surpreendentemente preservando acentuação. Trabalhos futuros devem lidar com o problema das fórmulas matemáticas.

O tamanho final do *dataset* em conteúdo textual é de 72 Gigabytes.

4.2 Estatísticas

As teses/dissertações estão parcialmente anotadas pelo meta-dados do dados abertos da CAPES. Umas das informações de maior interesse contidas nos metadados são as Grande Áreas de conhecimento e Áreas de conhecimento. Segundo as descrições das teses e dissertações nos dados abertos da capes[3] existem 9 grande áreas de conhecimento, que dividem todo conhecimento acadêmico, e são subdividias em áreas de conhecimentos. Essas áreas categorizam cursos, trabalhos técnicos, profissionais, bem como as próprias teses e dissertações.

Como se observa na Figura 4.3, as áreas com maior volume no corpus são as das ciências humanas e ciências sociais aplicadas, seguidas de multidisciplinar, saúde, engenharias, agrárias, exatas, com linguística, letras e artes e biológicas com menor volume. Também verificando a Figura 4.4, é possível analisar a distribuição entre as sub-classes (área de conhecimento) das grande áreas. Ciências Agrárias I tem grande prevalência na área de Ciências Agrárias, enquanto que educação se destaca nas ciências humanas, e interdisciplinar na área multidisciplinar. As demais áreas tem distribuições mais balanceadas, com destaque para administração pública em ciências sociais, e letras em linguística.

A Tabela 4.1 mostra os top 5 idiomas do corpus, evidenciando que a maior parte do conteúdo esta em português, com uma pequena parcela em inglês, e outras menores em diversas outras línguas.

O vocabulário de palavras diferentes foi contabilizados como 1.423 milhões de *tokens* diferentes. É necessário frisar, no entanto, que ainda restaram palavras com possíveis erros gramaticais, ou até sem significado semântico, como “aasds” ou “xxxcv”. Elas são, no entanto, pouco frequentes, e uma análise visual demonstra que uma simples filtragem por frequência reduz majoritariamente esses casos.

Os meta-dados do corpus oferecem mais informações de interesse. Em uma delas, foi possível calcular o número de teses/dissertações presentes

[3] `https://dadosabertos.capes.gov.br/dataset/2018-catalogo-de-teses-e-dissertacoes-da-capes`

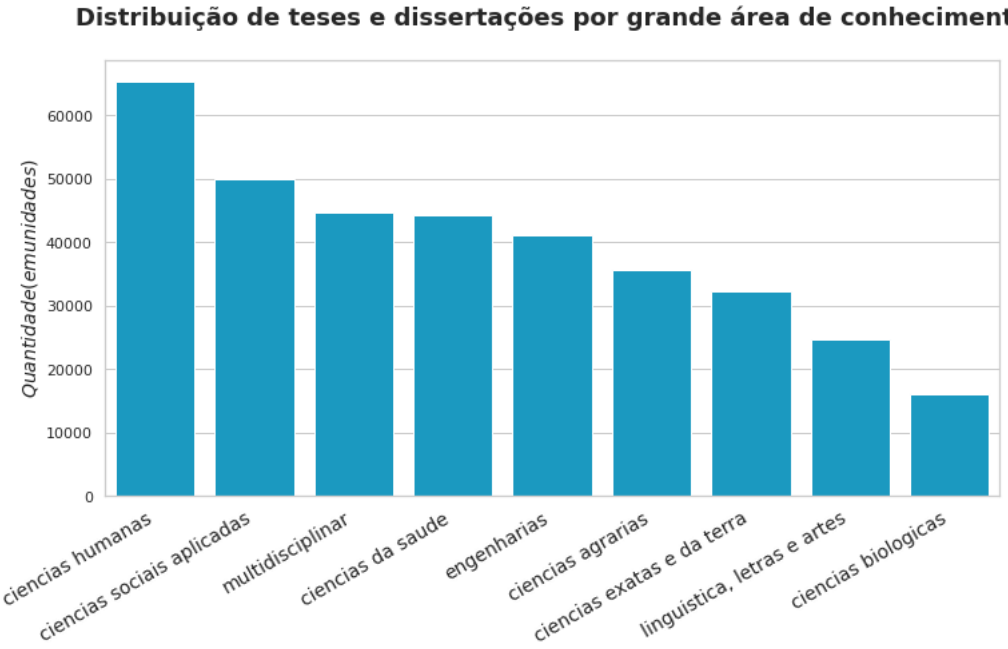


Figura 4.3: Distribuição de trabalhos de acordo com a grande área de avaliação disponibilizado nos dados abertos da plataforma Sucupira referente ao período de 2013-2018.

Tabela 4.1: Lista contendo os top 5 idiomas das teses e dissertações.

Idioma	Quantidade	Proporção
Português	346452	97,836%
Inglês	7100	2,005%
Espanhol	407	0,115%
Francês	55	0,016%
Outros	100	0,028%

no corpus por ano com links disponibilizando para download, mostrados na Figura 4.5. Percebe-se uma crescente disponibilidade de acesso aos trabalhos acadêmicos na plataforma Sucupira.

Também pode-se observar a distribuição de teses e dissertações do corpus por região, na Figura 4.6 com uma vantagem significante no sudeste para as demais. A Figura 4.7 mostra a distribuição por grau escolar, com dissertações de mestrado acadêmico sendo a mais frequente, e não havendo dissertações de doutorado profissional presentes.

Por fim, na Tabela 4.2 pode-se visualizar uma comparação do corpus proposto neste trabalho com os existentes na literatura. Em número de *tokens*, o corpus proposto ultrapassou o OSCAR, se tornando, pelo que sabemos, o maior corpus em língua portuguesa/Br. É importante destacar que o nosso corpus se diferencia do OSCAR e BrWaC já que a fonte deles vem de arquivos da internet, enquanto o aqui proposto vem de fontes acadêmicas.

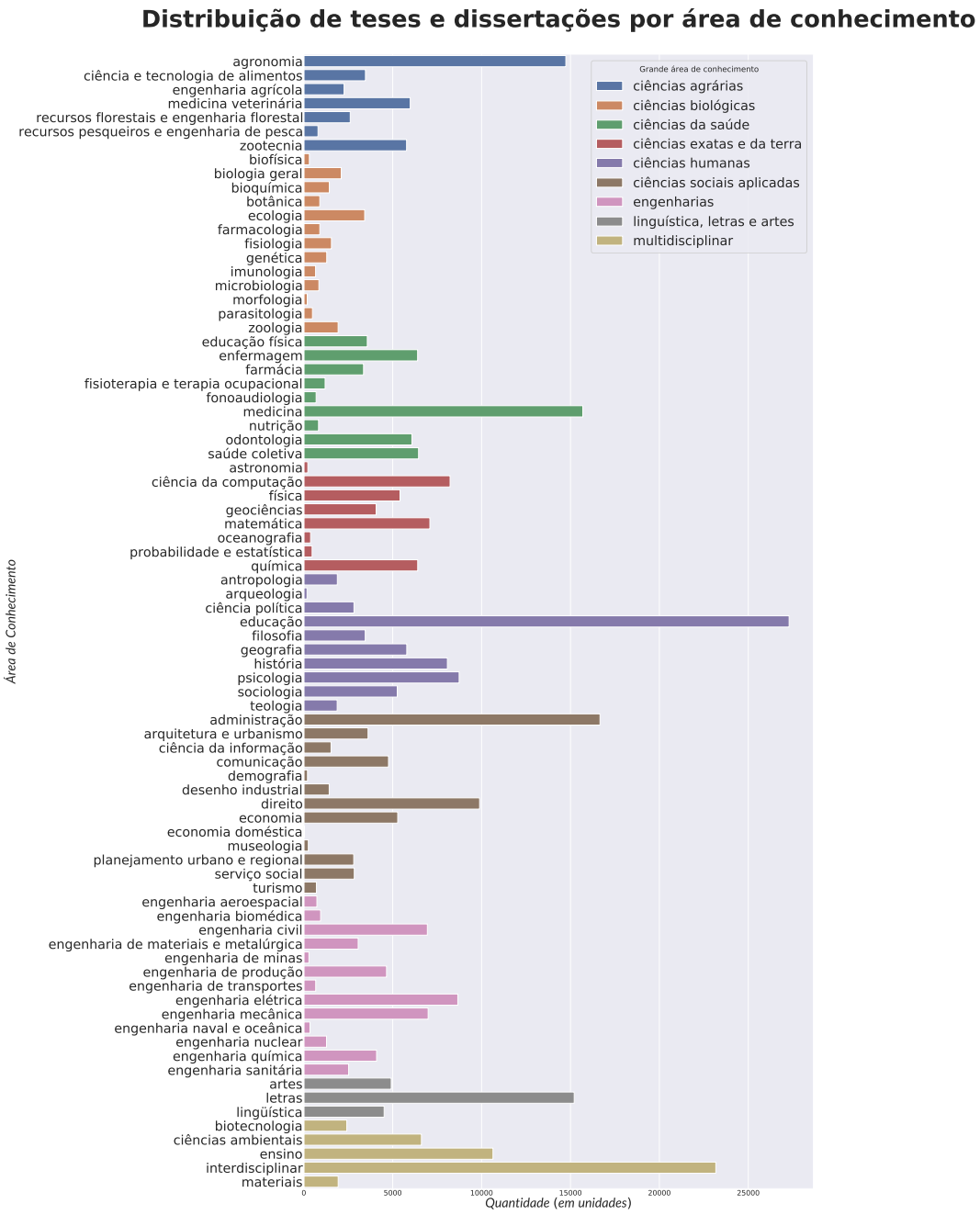


Figura 4.4: Distribuição das Áreas de Conhecimento nas Grande Áreas

Figura 4.5: Teses e Dissertações no corpus por Ano

Tabela 4.2: Lista contendo corpus em português com número de tokens. Destacados em cinzas são os corpus acadêmicos/científicos

Nome Corpus	**Fonte**	**Tokens**
BrWac	Arquivos Internet	2.1 bilhões
OSCAR	Arquivos Internet	10 bilhões
Wikipedia Corpus	Wikipédia	201 milhões
[12]	Dissertações	53 mil
[7]	Artigos Científicos	500 mil
[6]	Artigos Científicos	18 milhões
[13]	Artigos Científicos	5 milhões
[13]	Teses e Dissertações	34 milhões
Nosso Corpus	**Teses e Dissertações**	**11.6 bilhões**

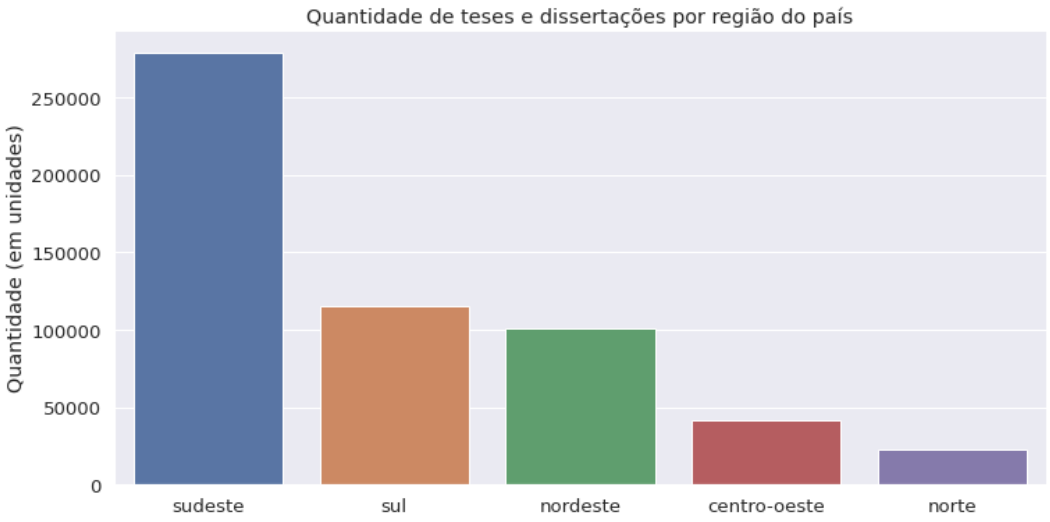


Figura 4.6: Teses e Dissertações no corpus por Região

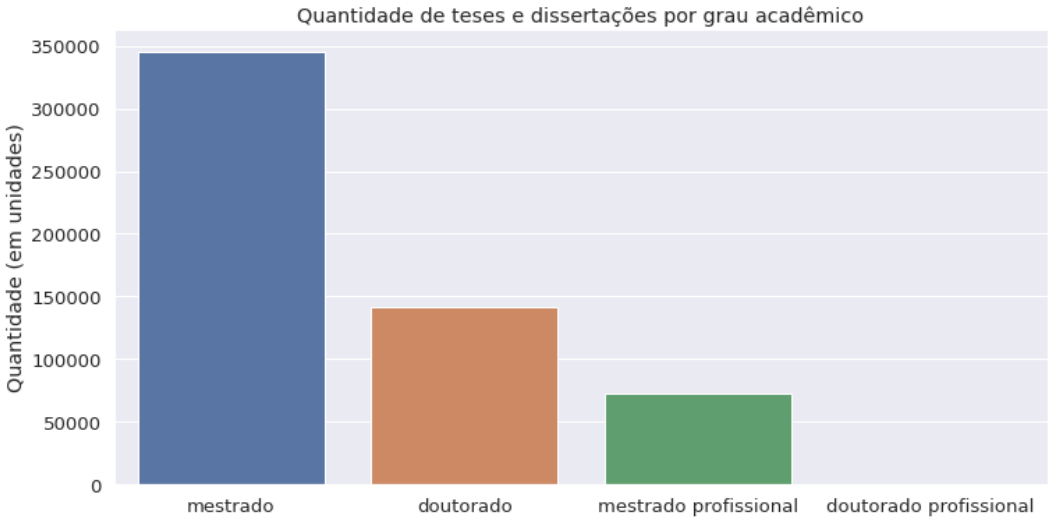


Figura 4.7: Teses e Dissertações no corpus por Grau

5
Construção de Modelos

Neste capítulo, aborda-se a construção de modelos na seguinte divisão: Modelos de linguagem que desconsideram o contexto, e modelos de linguagem que consideram o contexto. Avaliamos também os modelos treinados em tarefas de classificação.

5.1
Treinamento dos modelos

Para construção do modelo *bag-of-words* de *tfidf* foi construído um vocabulário filtrado com 1 milhão das palavras mais frequentes. Foi utilizada uma representação esparsa das matrizes, já que o tamanho dos vetores ultrapassavam a memória limite. A conversão em forma vetorial dos dados é feita somente na etapa de atualização dos algoritmos na tarefa em questão.

Para os modelos de vetor de palavra *word2vec* e *fasttext* um pré-processamento básico foi utilizado: as palavras do corpus foram transformadas no diminutivo e retirada toda acentuação e símbolos fora do alfabeto. Ambos foram treinados com uma representação de 300 dimensões, na versão skip-gram, essa sendo a versão que geralmente tem melhores resultados [42]. O treinamento foi feito com 16 núcleos, se passando 10 épocas, em cerca de 1 semana para cada, com taxa de treinamento 0.025, decaindo linearmente para 0.0001,

Para o modelo *glove* foi realizado o mesmo pré-processamento do *word2vec* e *fasttext*.

O modelo BERT for treinado com as seguintes configurações: Foi escolhido um vocabulário do *wordpiece* de 30 mil *tokens*. Os textos do corpus não foram pré-processados, mantendo-se assim palavras em sua forma capitalizada e com acentuação. Foi utilizada a implementação original em *tensorflow*[1]. O modelo do BERT escolhido foi o BERT Base com 110 milhões de parâmetros e 12 camadas, considerando-se os limites computacionais de espaço e tempo. Foi utilizada uma tpu v3-8 concedida pelo programa *TPU Research* do *google*. O modelo foi treinado com parâmetros similares ao do BERT original, por 1 milhão de passos, em sequencias de 512 *tokens*, e tamanho de *batch* 128.

[1] `https://github.com/google-research/bert`

Foi utilizado o otimizador adam com taxa de aprendizado 0.0001, $\beta_1 = 0.9$ e $\beta_2 = 0.999$, decaimento de pesos L2 de 0.01, uma taxa de aprendizado de aquecimento pelos primeiros 10 mil passos, com decaimento linear da taxa de aprendizado, e a probabilidade de mascarar *tokens* de 15%. O treinamento levou no total 1 semana. O modelo foi nomeado pela alcunha BERTAcadêmico.

5.2 Avaliações

Nessa seção avaliou-se os modelos de linguagem treinados no corpus acadêmico. São propostas 2 formas de avaliação: a por analogias, onde busca-se verificar se os vetores de palavras dos modelos estão condizentes com *benchmarks* já pré estabelecidos na literatura, e a tarefa de classificação, onde propõe-se um *dataset* de *abstracts* para categorização de macro e micro classes da área de conhecimento.

5.2.1 Analogias

Esta avaliação visa verificar a performance dos vetores de palavras em tarefas de analogias sintáticas e semânticas. O objetivo é averiguar como vetores de palavras treinados em um corpus de domínio acadêmico lidam com testes de conhecimento geral. Os testes foram propostos por [11] com as de analogia semântica contendo relações sobre:

- capitais comuns e países
- todas as capitais e países
- moeda e países
- cidades e estados
- relações de famílias

e como forma de analogia sintática, contém relações sobre:

- adjetivos e advérbios
- adjetivos opostos
- adjetivos bases e comparativos
- adjetivos bases e superlativos
- verbos no infinitivo e particípio presente
- adjetivos de países e nacionalidades
- verbos no infinitivo e formas do passado simples

Tabela 5.1: Analogias

Modelo	sintático	semântico
word2vec(LX-Corpus)	35.6	21.3
word2vec (Teses e Dissertações)	**42.3**	**21.5**
fasttext (LX-Corpus)	**58.7**	20.0
fastext (Teses e Dissertações)	57.1	**20.3**
glove (LX-Corpus)	45.9	**42.3**
glove (Teses e Dissertações)	**61.2**	39.7

– substantivos no singular e plural

– verbos no singular e plural

Para essas tarefas foram avaliados os modelos de vetor de palavra *word2vec* e *fasttext*. Os modelos de vetor de palavra de melhores resultados do LX-Corpus [72] foram colocados como comparação. A Tabela 5.1 mostra os resultados obtidos. Surpreendentemente, o modelo *word2vec* (teses/dissertações) apresenta melhores resultados na avaliação sintática que o *word2vec*(LX-Corpus), e um resultado próximo na avaliação semântica. Já o *fasttext* (teses/dissertações) apresenta uma leve piora em comparação ao *fasttext* (LX-corpus) na analogia sintática, com resultado aproximado na analogia semântico. O modelo com melhor performance balanceado é o *glove* treinado nas teses e dissertações, que obteve os melhores resultados na analogia sintática, enquanto que perde um pouco na analogia semântica para o *glove* (LX-Corpus).

5.2.2 Classificação de resumos

Para avaliar os modelos na sua especialidade, é necessário um *dataset* que possua conhecimento acadêmico científico. Essa é uma limitação, pois mesmo em inglês existem poucos *datasets* anotados deste tipo. Foi decidido então construir um *dataset* por meio dos metadados disponíveis dos dados abertos nas dissertações. A riqueza em detalhes das teses e dissertações permitiram a construção de tarefas de classificação desafiadoras, as quais serão detalhadas a seguir.

Em ambas as tarefas, foram selecionadas todos os resumos das teses/dissertações com uma das 9 grande-áreas de conhecimento definidas, de 1987 a 2019, que não estavam contidos entre as 392 mil que constituem o corpus, totalizando 824 mil resumos no total. Para a primeira tarefa de classificação, foram selecionadas todos os resumos das teses/dissertações com uma das 9 grande-áreas de conhecimento definidas, sendo que apenas 8 das 392 mil

não continham esse *label*. Para a segunda tarefa, foi classificada em área de conhecimento, sendo elas sub áreas das grandes áreas. Dos 824 mil resumos, 421mil não continham o *label* da área de conhecimento, contabilizando um *dataset* com tamanho total de 403 mil resumos.

Foram avaliados o modelo *tfidf*, os modelos de vetores de palavras estáticos *word2vec*, *fastext* e *glove*, e o modelo contextual BERTAcadêmico. Com o propósito de comparação, foi adicionado também o modelo de linguagem contextual estado-da-arte, o BERTimbau[53], e também o *glove* do LX-corpus.

O *dataset* foi dividido em treino, teste e validação, nas proporções 70%, 15% e 15%. Para a classificação das Grande Áreas de Conhecimento, tem-se um total de 576.800 resumos no treino, e 123.600 na validação e teste.

Para classificar com o *TF-IDF*, se extrai a representação em *bag-of-words* para o classificador, uma rede neural *totalmente conectada* de 2 camadas com 200 unidades cada, seguida por uma camada não linear de *softmax*.

Para o *word2vec*, *fasttext* e *glove*, as matrizes dos vetores de palavras são incorporados a uma arquitetura de rede convolucional com a seguinte configuração: 3 camadas convolucionais 1D de tamanho 128 com ativações *ReLu*, onde as duas primeiras são acompanhadas de max pooling 1D de tamanho 5, e a ultima de Global *max pooling* 1D, seguida de um camada *totalmente conectada* de tamanho 128, e a camada *softmax* para classificação. O *dataset* é processado com truncação de limite máximo em 512 palavras, e *padding* com zeros são utilizado para preencher os resumos menores que o limite. Foi utilizado o otimizador *adam* com taxa de aprendizado *0,001*.

Essa é uma rede mais elaborada que a do *tfidf* pelo fato que abordagens *bag-of-words* são representações com mais linearidade que os vetores de palavras, e, como foi testado na prática, não melhoram a performance do *tfidf*.

No BERTAcadêmico e BERTimbau foram abordadas a estratégia de *finetuning*: todas as camadas são reajustadas a nova tarefa de classificação, treinadas com o otimizador Adam com taxa de aprendizado de *0,0001* com o tamanho de *batch* 16 por 3 épocas.

Como forma de tentar se obter uma avaliação justa entre modelos, todas utilizam pré-processamento básicos: nos modelos de *tfidf*, *word2vec*, *fasttext* e *glove* as palavras são transformadas nos seus diminutivos, retira-se acentuação e números. Para os modelos contextuais, mantemos o texto na sua forma original, já que eles foram treinados com textos do corpus nessa forma. Não foram realizadas buscas extensas nos hiper-parâmetros.

A média ponderada da precisão, *recall* e *f1* é demonstrado na Tabela 5.2. O pior modelo foi o que faz uso dos vetores de palavras do *glove*(LX-Corpus). Surpreendentemente, o modelo *tf-idf* possui resultados competitivos com o

Tabela 5.2: Grande Área de Conhecimento

Modelo	*precision*	*recall*	*f1*
glove(LX-Corpus)	0.64	0.63	0.62
tfidf	0.74	0.75	0.75
word2vec	0.72	0.75	0.74
fastext	0.76	0.74	0.75
glove	0.75	0.75	0.75
BERTimbau	0.74	0.75	0.74
BERTAcadêmico	**0.76**	**0.77**	**0.77**

Tabela 5.3: F1 por Grande Área de Conhecimento

Grande Área	glove	tfidf	word2vec	ftt	BERTAcademico	Bertimbau
Ciências Agrarias	0.83	0.83	0.83	0.83	**0.85**	0.81
Ciências Biológicas	0.73	0.72	0.73	0.73	**0.75**	0.67
Ciências da saúde	0.83	0.83	0.83	0.84	**0.85**	0.82
Ciências exatas e da terra	0.78	0.78	0.78	0.79	**0.81**	0.76
Ciências humanas	0.79	0.78	0.79	0.79	**0.81**	0.79
Ciências sociais	0.78	0.77	0.77	0.77	**0.78**	0.78
Engenharias	0.77	0.77	0.77	0.77	**0.77**	0.75
Linguística, letras e artes	0.85	0.85	0.85	0.86	**0.88**	0.85
Multidisciplinar	0.34	0.33	0.32	0.32	**0.36**	0.33

demais modelos neurais, com somente o BERTAcadêmico possuindo uma leve vantagem. Os modelos *word2vec*, *fasttext* e *glove* tiveram performance similares, e o modelo contextual BERTimbau obteve um bom resultado, condizente com a maioria dos modelos, considerando que foi pré treinado em conteúdo possivelmente não acadêmico(*blogs*). Resultados mais detalhados das Grande Áreas de Conhecimento estão na tabela 5.3.

As classes com melhores resultados em todos os modelo são as de linguísticas, saúde e agrárias, seguidas de exatas, sociais e engenharias e biológicas. A Classe multidisciplinar é a de mais difícil classificação, a qual pode possivelmente ser atribuído ao fato que ela engloba teses/dissertações de múltiplos conteúdos acadêmicos das outras áreas. Como no resultado anterior, o BERTAcadêmico obteve a melhor performance.

O resultado da avaliação na área de conhecimento esta na Tabela 7.2. Todos as configurações de treinamento de todos os modelos são mantidas, bem como a mesma proporção de particionamento do *dataset*, sendo 302 mil para treino, 60 mil para validação e teste.

Observa-se uma proximidade na performance de todos os modelos, com um ganho pequeno do BERTAcadêmico sob os demais. Observando o resultado mais detalhado das micro classes no apêndice, vemos que o ganho do BERTA-

Tabela 5.4: Área de Conhecimento

Modelo	*precision*	*recall*	*f1*
glove(LX)	0.36	0.40	0.35
tfidf	0.62	0.61	0.61
word2vec	0.61	0.60	0.60
fasttext	0.62	0.61	0.61
BERTimbau	0.58	0.60	0.58
BERTAcadêmico	**0.63**	**0.64**	**0.63**

cadêmico se concentra principalmente nas classes com pior performance nas demais.

5.3 Análise dos resultados e Discussão

As tarefas de analogia tinham como objetivo averiguar se os vetores de palavras gerados no corpus possuem e retém informações para responder questões de conhecimento gerais. Os resultado foram promissores, indicando que os vetores de palavras podem possivelmente ser usados para outras tarefas além do domínio acadêmico .

As tarefas de classificação com base nas macro e micro classes da área de conhecimento se demonstraram mais desafiadoras do que imaginadas inicialmente. Embora haja um ganho ao se utilizar modelos contextuais, os resultados ainda estão longe de serem promissores.

É necessário frisar no entanto que, não foi realizada nenhuma outra estratégia específica para melhorar resultados, como busca exaustiva de hiperparâmetros [54], *data augmentation*[55], ou pré-processamentos mais elaborados. O objetivo desta avaliação é somente fazer um *benchmarking* inicial dos modelos em condições de maior igualdade possível, para esta tarefa que, pelo conhecimento do autor, é única em português.

Por fim, os resultados do BERTimbau abaixo dos demais modelos é um fator curioso que demanda uma investigação mais profunda além da simples análise dos resultado. Uma linha interessante de investigação é o vocabulário do BERTimbau em comparação ao BERTAcadêmico. Ambos possuem um vocabulário de 30 mil *tokens* e utilizam o mesmo *tokenizer*, o *Wordpiece*, mas a intersecção de *tokens* entre eles é de somente 31%. Em comparação, o modelo de domínio científico em inglês *SciBERT* possuem uma intersecção de 42%. Esse fato nos diz que existe uma diferença maior entre os vocabulários dos modelos em português do que os em inglês. Esse pode ser um fator possivelmente impactante nos resultados.

6
Análise Semântica do Corpus

O objetivo deste capítulo é a tentativa de análise semântica do *corpus*, com ajuda dos modelos e de técnicas que permitam visualização de informações de interesse. Para tal, duas abordagens serão tomadas: a visualização dos vetores de palavras das teses/dissertações com dimensionalidade reduzida pelo UMAP, e a visualização de tópicos intrínsecos ao corpus, por meio de um modelo de tópicos, o LDA.

6.0.1
Visualização

Foi utilizado os resumos de teses e dissertações por serem mais representativos (1.2 mil resumos contra 392 mil textos completos) e para uma experimentação mais rápida dos parâmetros. Como pré-processamento foram colocadas todas as palavras no diminutivo e retirada a acentuação. Foi utilizada a lista de *stopwords* do NTLK[59] que retira termos comuns da linguagem, que não são de interesse. Um modelo de n-gram, com n=2 é treinado para detecção de frases, juntando palavras com o caractere "_".

Na tabela 6.1 vemos os 10 tópicos mais relevantes considerados pelo LDA, em ordem decrescente. O primeiro e mais relevante tópico parece conter termos sobre a sociedade e o valores sociais humanos. O segundo tópico contém termos sobre a educação e pedagogia. O terceiro tópico contém termos sobre saúde e família, além de consultas e coleta de dados. O quarto tópico parece capturar termos comuns que aparecem em resumos de teses e dissertações: projetos, propostas, métodos etc. O quinto tópico possui termos sobre politica e politicas, organização nacional (estado, cidades) e assuntos relacionados. O sexto tópico contém termos ligados a literatura e artes. O sétimo tópico possui termos sobre experimentos e experimentos em seres vivos(ratos primariamente). O oitavo tópico prece ter termos da física e matemática. O nono tópico contém termos da área de direito e similares. O décimo e ultimo tópico possui termos relacionado as áreas de informática e tecnologia.

Tabela 6.1: Tópicos

Tópico	**Termos**
1	sociedade, sentido, discurso, social, sujeito
2	educacao, escola, formacao, professores, ensino
3	saude, profissionais, cuidado, familia, enfermagem
4	desenvolvimento, projeto, metodologia, design, estudos
5	politicas, politica, politicas_publicas, cidade, social
6	obra, arte, romance, narrativa, personagens
7	animais, ratos, induzida, tratamento, efeitos
8	escoamento, modelo, simulacao, metodo_elementos, elementos_finitos
9	direito, direitos, lei, justica, juridica
10	software, redes, sistemas, arquitetura, aplicacoes

No anexo esta presente uma listagem com mais termos. Uma visualização dinâmica com os top-50 tópicos está disponível em [1]

6.1 Visualização dos Embeddings

A visualização de dados por meio da clusterização permite identificar como se organizam e até como se relacionam diferentes grupos de dados. Mas para conseguir uma visualização relevante, é necessário fazer a redução de dimensionalidade até um nível intuitivo ao humano: o de 2 dimensões. Uma redução tão extrema exige algoritmos especializados, que vão além das técnicas padrões. Um desses algoritmos é o UMAP, que será explorado neste trabalho.

6.1.1 Visualização

Para a criação da imagem mais representativa das teses/dissertações brasileiras conhecida, optou-se por fazer uso dos resumos contidos nos metadados, já que tem-se a limitação de somente 392 mil trabalhos com o texto completo, diante dos 1.2 milhões de resumos. Outras motivações para tal são as questões de limite computacionais de tempo e espaço para lidar com documentos inteiros de teses/dissertações, o que pode ser tratado em trabalhos futuros.

Para extração dos vetores de palavras foi utilizado o BERTAcadêmico. Para cada resumo das Teses e dissertações, foram extraídos os vetores das palavras e calculado o vetor da média, tendo assim um único vetor para cada

[1] `https://academicai.vercel.app/projects/graphics`

resumo. Dos 1.2 milhões de vetores, 300 mil foram utilizados para treinar as rede do UMAP, com um *batch* de tamanho 64 e otimizador adam. Os vetores tiveram sua dimensionalidade reduzida a 2 pelo UMAP.

A imagem resultante é mostrada na Figura 6.1. As teses/dissertações foram coloridas com suas respectivas grande áreas de conhecimento.

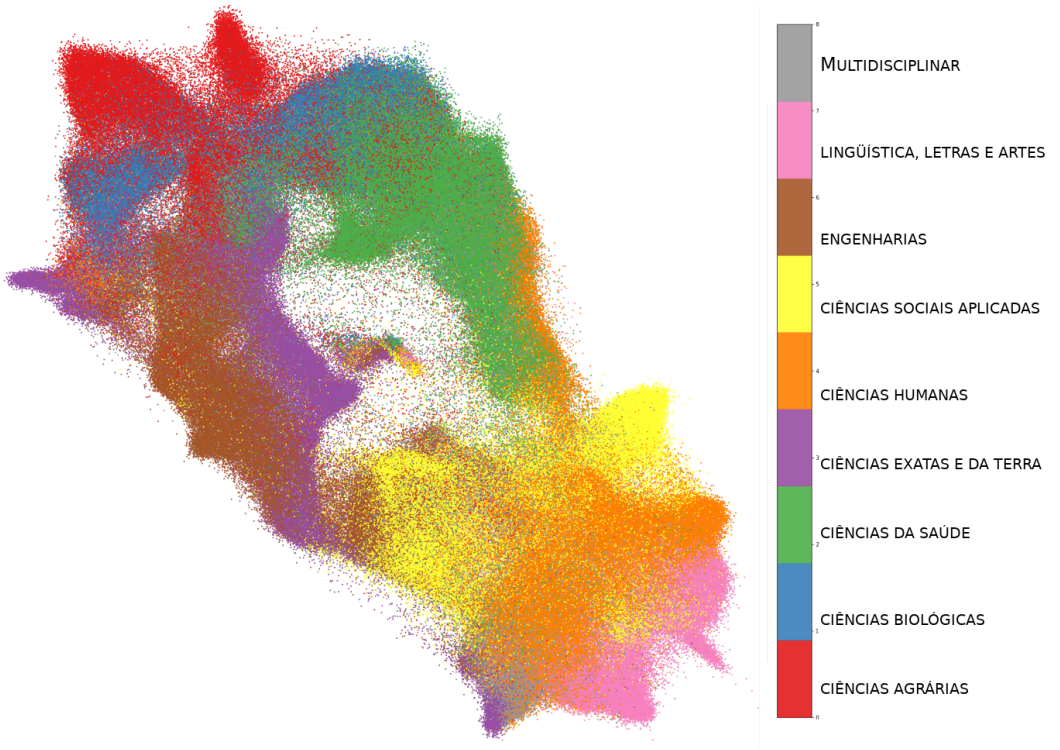


Figura 6.1: Visualização dos Vetores dos Resumos

Todas as grande áreas possuem um agrupamento significante o bastante para permitir identifica-las de imediato, com exceção da área multidisciplinar. Ao examinar mais profundamente, essa área está contida em todos os demais agrupamentos, com destaque a intersecção entre as áreas de Letras, Linguagem e Artes (LLA), humanas e um subgrupo de exatas.

É perceptível também a proximidade entre os agrupamentos de LLA, humanas e sociais, áreas que parecem ter bastante interseções. A área de sociais parece ter interseções consideráveis com a área de humana, que a divide em dois corpos concentrado, e um pequeno mais próximo a LLA. A área de humanas também tem uma pequena faixa de presença na área de ciências da saúde. LLA por sua vez, é isolado abaixo a direita na imagem, tendo maior intersecção com humanas.

A área de exatas aparenta ser, depois da multidisciplinar, a mais dispersa. No entanto, é possível identificar claramente seus subgrupos, localizados em três pontos: um grande corpo ao lado da área de engenharia, outro grupo mais a esquerda próximo de biológicas e engenharia, e um ultimo grupo mais distante,

próximo as áreas de LLA, sociais e Multidisciplinar. Uma possível interpretação é de que exatas aparenta ser também um campo multidisciplinar, com bastante uso nas demais áreas com quem faz fronteira.

A área de engenharia possui uma forte concentração a esquerda na imagem, com uma proximidade muito grande com a área de exatas, indicado alguma correlação entre as áreas. Ela também tem uma notável proximidade e presença na área de sociais.

A área de ciências da saúde está presenta a direita na imagem com uma interseção mais notável em relação a área de biológicas. Essa por sua vez, é bem dividida e se mistura com várias áreas, dentre as mais notável as área de agrarias e saúde, mas parecendo também ter intersecções com engenharias e exatas. Por ultimo, a área de agrárias esta localizada acime e a direita na imagem, divida em dois grupos que fazem fronteira com a área da saúde e uma faixa que se intersecta com múltiplas áreas.

Por fim, é interessante notar a ilha presente no centro da imagem: Ela parece ser uma área de incerteza, com representantes de diversas áreas reunidos.

6.2 Discussão

Nesse capítulo foram descritas abordagens utilizadas na exploração do *corpus*, além de estatísticas básicas como as apresentadas no Capítulo 4. A primeira abordagem foi a de modelagem de tópicos. Nela, tentou-se adentrar no *corpus* para responder uma pergunta simples, mas com resposta difícil de se obter: "Sobre o que trata o *corpus*?". De forma bem simplificada, o LDA tenta descobrir isso ao considerar que existem variáveis latentes (que queremos descobrir) das quais, por meio de um processo gerador, podem produzir os próprios documentos(os quais conhecemos). Os resultados foram satisfatórios, visto que há pouca correlação entre os tópicos descobertos.

E como é comum na ciência, uma pergunta simples pode sempre levar a mais perguntas. "Em um corpus tão grande não pode haver uma miríade de tópicos relevantes?", "Como os tópicos se relacionam?", "Existem hierarquias de tópicos?", dentre muitas outras questões. Para algumas delas existem caminhos interessantes a se explorar: há métodos que tentam descobrir um número de tópicos ideal automaticamente[62]; existem variações do LDA que constroem grafos e árvores de tópicos [63][64]. Trabalho futuros devem abordar essas e outras técnicas.

A segunda abordagem considerou olhar para o corpus através das lentes da clusterização. Encontrar grupos de interesse, com quem cada um se

aproxima mais, pode ser um valioso *insight* para pesquisas mais minuciosas e aplicações que fazem uso de vetores de palavras acadêmicos. A visualização no entanto, ainda é superficial. Não há garantias que esses são os melhores vetores de palavras para clusterização, e nem que o UMAP é o melhor método de clusterização. Ele é no entanto, pelo que foi pesquisado, o com melhor performance/desempenho para *datasets* grandes, devido a abordagem de redes neurais que permite treinamentos em *batches*.

Além desses fatos, somente as Grande Áreas de conhecimento foram tocadas. Existem mais de 80 áreas de conhecimento, que podem ser consideradas micro-categorias. Trabalhos que exploram essa linha já estão em andamento.

Por fim, existem várias outras abordagens para se explorar o corpus acadêmico coletado. O uso e construção de ontologias[66], taxonomias[65], dentre outras formas de se organizar e inferir conhecimento, são linhas promissoras e de relevância para a ciência.

7
Aplicações

Este capítulo propõe uma forma de aplicação de caso real dos modelos treinados no corpus acadêmico, apresentando a ferramenta de pesquisa textual com um QA Interativo (IQA): o SucupiraBot[1][1]. São feitas avaliações das performances dos modelos, bem como do sistema como um todo.

7.1
Plataforma Sucupira

A plataforma Sucupira é a principal fonte de dados da pós graduação no Brasil, sendo constituída de informações diretamente relatadas pelos diversos programas de pós graduação públicas e privadas do país.[2] Para este trabalho, foram selecionados 3 conjuntos de dados disponíveis pela plataforma, que são: (i) produções intelectuais (selecionamos artigos de periódicos e Anais), (ii) trabalho de conclusão de curso (teses e dissertações), e (iii) periódicos (nacionais e internacionais). Adicionalmente, para simplificar a versão inicial do *IQA*, foram selecionados as propriedades desses dados considerados de maior uso.

7.2
Construção do dataset

Para construção do *dataset* se escolheu utilizar a linguagem específica de domínio *Chatette*[3], muito utilizada em conjunto com o Rasa. Ela funciona como um gerador de linguagem natural por meio de *templates* definidos pelo usuário, com possibilidade de atribuir operadores estocásticos que permitem a construção rápida e eficiente de vários exemplos a partir de um único *template*.

As entidades definidas pelo modelo são compostas pelas propriedades da Figura 7.1. Em uma pergunta como "Artigos da área de química", há as entidades "artigo" e "área", com valores "Artigos" e "química" respectivamente.

Para as intenções, foram definidas três categorias. São elas:

[1] `https://academicai.vercel.app/projects/sucupirabot`
[2] `https://www.capes.gov.br/avaliacao/plataforma-sucupira`
[3] `https://github.com/SimGus/Chatette/wiki`

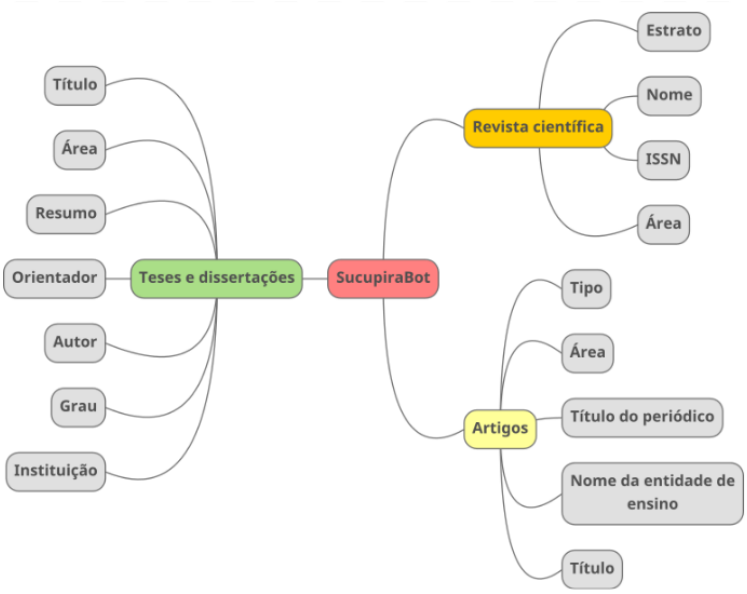


Figura 7.1: Dados da plataforma sucupira abordados pelo IQA.

- **Selecionar**: Essa intenção abrange questões de busca, onde o usuário tem que obrigatoriamente definir o tipo (Teses ou dissertações, Artigos, ou Revista), e uma área científica desejada (matemática, computação, etc).
- **Filtrar**: Essa intenção tem como objetivo diminuir o número de resultados obtidos de uma questão com a intenção anterior. Por exemplo: "me mostre apenas as teses da PUC-Rio".
- **Clarificar**: Essa intenção tem como objetivo a verificação individual de algum resultado, como por exemplo: "Qual o autor do artigo 3".

Para a tarefa de *Named-entity recognition* por causa da necessidade de uma quantidade significante de dados rotulados para treinar os modelos, optou-se por elaborar uma estratégia de criação de *dataset* sintético.

A abordagem tomada foi criar diversos *templates* de questões possíveis dos usuários do sistema, com as entidades sendo preenchidas pelos dados do sucupira. Essa abordagem é similar a [71], onde eles utilizaram *templates* para gerar um *dataset* sintético sobre uma base de conhecimento. No *dataset* do *SucupiraBot* existem duas categorias de entidades: as entidades dos dados, que incluem todos os dados descritos na Figura 7.1, e a entidade identificando termos de busca. Para o segundo, sentenças são retirados de títulos e resumos. Exemplos de *templates* são mostrados na Tabela 7.1, com as duas categorias representadas por $< a >$ e $< b >$.

Por último, é importante destacar que por ser uma abordagem de geração sintática de dados, é muito fácil de sobre-ajustar os modelos caso as perguntas

Tabela 7.1: Templates

Templates	Instâncias	Intenção
Quero **<a>** sobre **<b>**	Quero **artigos** sobre **k-complexos em sinais eeg**	Seleção
Acerca de **<b>** gostaria de **<a>**	Acerca de **k-complexos em sinais eeg** quero **artigos**.	Seleção
Sobre o último resultado, quero saber **<b>**	Sobre o último resultado, quero saber o **autor**	Clarificação

sejam similares. Para evitar isso foram adotadas duas estratégias: ao ser escrever o *template*, os fazer com a maior diversidade no tempo disponível, e utilizar ruído nos dados gerados na hora de treinar. O ruído em dados textuais são ações como trocar, inserir e deletar caracteres e palavras.

7.3 Sistema

O *IQASucupiraAPI* desenvolvido permite fazer requisições *http* utilizando os verbos *GET* e *POST*. Dessa forma, qualquer tecnologia que tenha suporte a essas operações pode se comunicar com nosso IQA, facilitando a interoperabilidade e uso de nosso sistema.

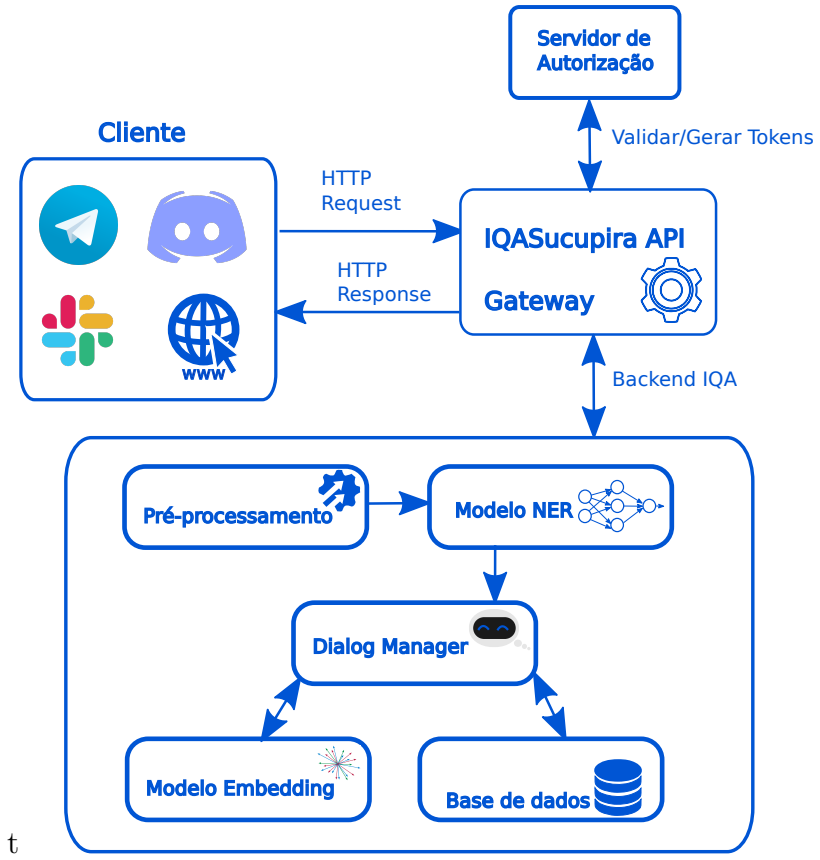
Para utilizar o *endpoint* de consulta ao IQA, é necessário alguns requerimentos, como estar registrado no IQASucupiraAPI. No primeiro acesso o cliente necessita se cadastrar, informando um email e senha que serão utilizados para obter o *Token* de acesso que é solicitado nos demais *endpoints* da aplicação. *Tokens* de acesso e de atualização são fornecidos de acordo com as especificações do *JSON Web Tokens*.[4]

Foi desenvolvida interface conversacional acessível via *browser* com o *framework Next.js* como ilustrado na Figura 7.3, além de *chatbots* para *Discord* e *Telegram*.

O pipeline do servidor, ilustrado na Figura 7.2, se dá na seguinte ordem: dada a entrada recebida em texto, ela é direcionada para o módulo de pré-processamento, que retira caracteres especiais e acentos, força a conversão para letras minúsculas e retira qualquer ocorrência de múltiplos espaços em branco. Uma cópia da entrada pré-processada é passada para o modelo de NER, que extrai as entidades da entrada.

Considerando o desempenho do sistema quanto a processar e calcular os vetores de palavras de centenas de milhares de resultados, optou-se por fazer

[4] `https://jwt.io/introduction/`



t

Figura 7.2: Visão geral do SucupiraBot.

o processamento de modo *offline*, salvando em uma *hash table* os vetores de palavras dos títulos dos trabalhos que podem ser pesquisados.

O *Dialog Manager* cuida então de todo processamento importante que se segue. Dada a identificação da intenção da pergunta, e no caso da seleção, das entidades dos termos de pesquisa ou de área, uma consulta pode ser realizada no banco de dados para retornar os resultados requiridos.

Por fim, o *Dialog Manager* extrai os vetores de palavras da sentença original, que são comparados com os vetores de palavras dos resultados, e os ordena pela similaridade com a sentença original.

7.4 Experimentos

Para esta avaliação, foram escolhidos os modelos de vetores de palavras treinandos no corpus acadêmico *fasttext* e *word2vec*,o modelo *glove* treinado no LX-Corpus e os *transformers* BERTAcadêmico e BERTimbau.

Acerca do *dataset* de entidades e intenções, foram escritos cerca de 300 *templates* com formas diferentes de realizar as perguntas, e dos quais foram instanciados 50 mil questões. Como forma de lidar com o *overfitting*, foi aplicado sobre este *dataset* operações de inserção de ruído. Para cada

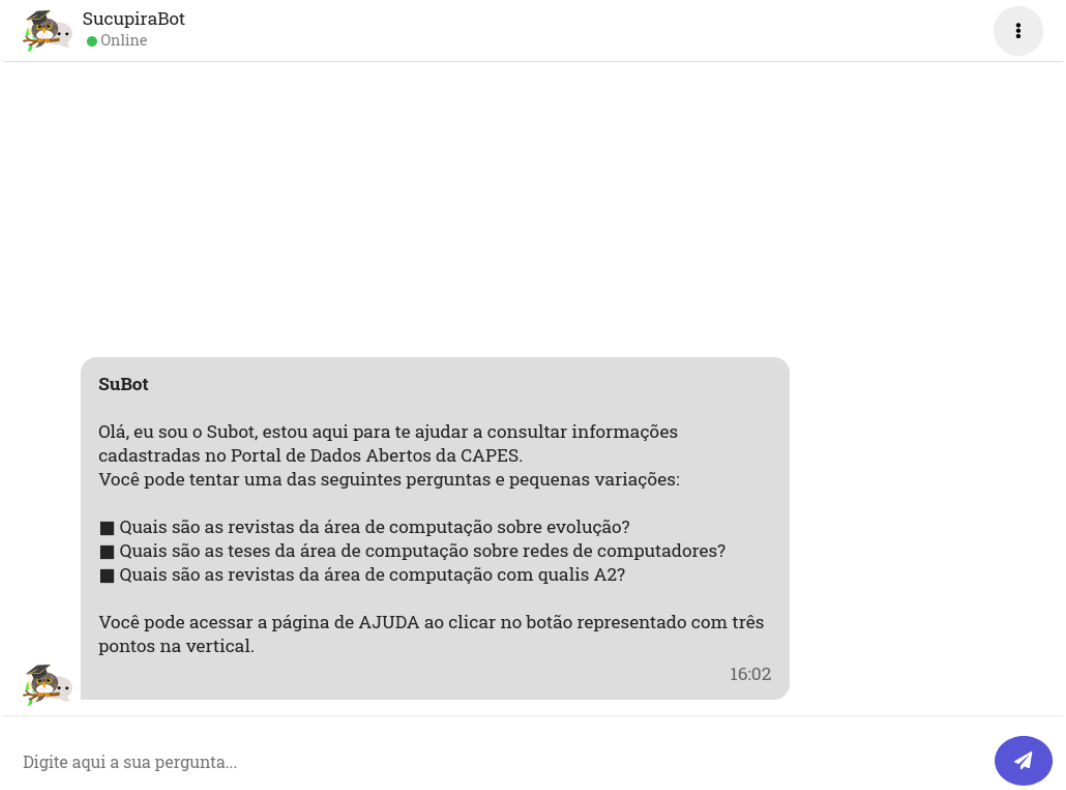


Figura 7.3: Versão *web* do SucupiraBot

Tabela 7.2: Micro-Classe

Modelo	precision	recall	f1
convNet	0.93	0.94	0.94
glove(LX-Corpus)	0.94	0.94	0.94
word2vec	0.95	0.94	0.95
fasttext	0.95	0.95	0.95
BERTimbau	0.96	0.95	0.96
BERTAcadêmico	**0.96**	**0.96**	**0.96**

palavra, tem-se a chance de 5% em ela ser trocada com outra palavra, deletada, substituída por uma palavra similar através de um dicionário de sinônimos, ou ser corrompida com a inserção de algum caractere.

7.4.1 Avaliação de Intenções

A avaliação de intenções se trata de classificar as questões exemplos em três classes: seleção, filtragem e clarificação. Para os modelos de vetores de palavras, foi utilizada a mesma rede neural convolucional da Subseção 5.2.2. Os modelos *transformers* também foram ajustados com as mesmas configurações da Subseção 5.2.2.

Como pode-se observar, não há muita diferença entre os modelos nesta

Tabela 7.3: NER

Modelo	precision	recall	f1
Bidirecional	0.89	0.88	0.88
glove(LX-Corpus)	0.91	0.92	0.92
word2vec	0.93	0.91	0.93
fasttext	0.95	0.92	0.94
BERTimbau	0.94	0.93	0.93
BERTAcadêmico	**0.94**	**0.95**	**0.95**

tarefa, com todos obtendo resultados próximos, acima de 95% de f1.

7.4.2 Avaliação NER

Para avaliação NER dos modelos com os vetores de palavras, foi utilizada uma rede bidirecional com as seguintes configurações: duas camadas de LSTM, com 128 unidades cada (que formam a bidirecional), seguida por múltiplas camadas de *softmax*, uma para cada sequência de saída da rede recorrente. O tamanho máximo da sequência de *tokens* foi definido como 30. No experimento, foi considerada uma versão do modelo com vetores de palavras inicializados aleatoriamente, e outras duas com os vetores de palavras do *word2vec* e *fasttext*.

Para os modelos *transformers*, só é necessário adaptar a última camada para o número de *tokens* definido, já que eles naturalmente suportam múltiplas entradas e saídas.

A média das métricas de *precision*, *recall* e *f1* para cada entidade é mostrado na Tabela 7.3

É notável que existe uma diferença ao se utilizar vetores de palavras treinados nos resultados: o modelo bidirecional inicializado com vetores de palavras aleatórios obteve um resultado com 4 pontos de *f1-score* abaixo do modelo de pior resultado utilizando *embedding* pré-treinados. Os melhores resultados foram obtidos pelos modelos BERTAcadêmico e BERTimbau, seguidos de perto pelo *word2vec* e *fasttext*.

7.4.3 Avaliação vetores de palavras

Como forma de avaliar a performance dos vetores de palavras, é feita uma comparação dos diferentes resultados obtidos através de uma consulta a títulos de teses/dissertações. A consulta é feita com a ordenação por meio da similaridade dos cossenos entre o *embedding* da pergunta e os vetores de palavras das teses/dissertações. Como exemplo, para a solicitação “dissertações

sobre diversidade de gênero" é ilustrado na Tabela 7.4 os cinco resultados mais similares a pergunta para cada modelo.

Nos resultados percebe-se uma tendência entre os modelos que entende a pergunta em um sentido biológico, sendo que ela é mais utilizada para o sentido social. O modelo *Glove* (LX-Corpus) retorna os resultados mais distantes, com somente talvez sua 2 resposta condizente com a consulta. O Modelo *FastText* treinado no corpus acadêmico tem seu *top-3* relacionado com o tema, seguido de temáticas sobre espécies e climas. O BERTimbau tem resultados misturados, com presença de resultados sobre ecologia e evolução. O BERTAcadêmico teve os resultados mais condizentes com a intenção buscada na consulta, com seu quarto resultado podendo ser mais questionado.

7.4.4 Avaliação questionário

Para avaliação humana do *SucupiraBot*, se realizou o estudo com 16 pessoas. Os participantes foram informados sobre o uso de seus dados de forma anonimizada nessa pesquisa acadêmica, com sua participação sendo totalmente voluntária. Uma distribuição por formação é mostrado na Figura 7.4.

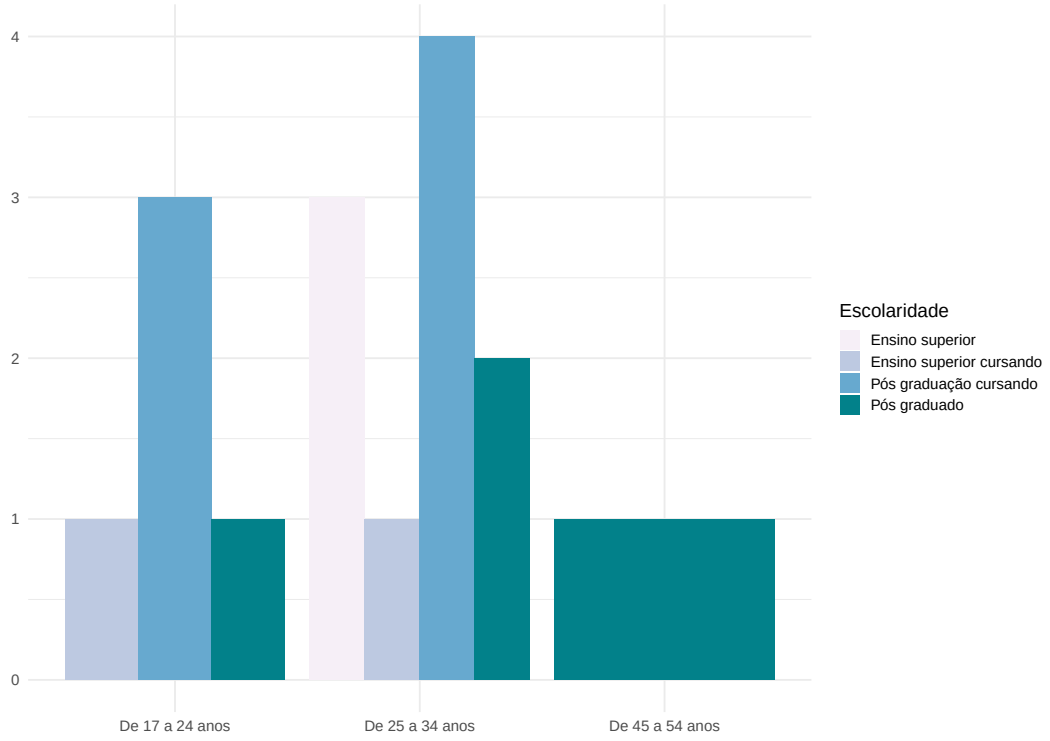


Figura 7.4: Sumário das respostas ao questionário.

Para o estudo, foi desenvolvido um roteiro contendo perguntas que abrangem todos os possíveis tipos de questões respondidas atualmente pelo *chatbot*, ordenadas das mais fáceis às mais difíceis. O roteiro é dividido em 6

Tabela 7.4: Top-5 títulos de dissertação mais similares

Glove(LX-Corpus)
Estudo exploratório de extratos aquosos de semente de uva e de chá verde para possível uso em desinfecção de águas residuárias.
Relação família-escola: processos de produção de subjetividades.
Produção de mudas de açaizeiro solteiro (euterpe precatória mart.) sob diferentes doses de nitrogênio e potássio.
Avaliação de programas de alimentação na preparação de marras de alto valor genético.
Validação de um protótipo de equipamento de avaliação macroscópica de modelos de atrito de hastes em regime de pre deslizamento.
FastText(Corpus-Acadêmico)
Experiências de gênero e formação de preferência de estudantes superdotadas
Relação família-escola: processos de produção de subjetividades.
As noções de masculino e feminino: concepções ideológicas e papeis de gênero.
Diversidade de mucoromyceta em brejos de altitude de pernambuco, brasil, e avaliação de espécimes produtores de proteases coagulantes do leite.
Influência do gradiente climático na riqueza de especies e número de modos reprodutivos de anuros amazônicos.
BERTtimbau
Diversidade, imigração e desenvolvimento.
Influência do gradiente climático na riqueza de espécies e número de modos reprodutivos de anuros amazônicos.
Experiências de gênero e formação de preferência de estudantes superdotadas.
Abundância, densidade, padrões de atividade e ecologia espacial de carnívoros no parque estadual do Rio Doce - MG.
Jogos evolucionários sobre grafos estrela fechada.
BERTAcadêmico
Experiências de gênero e formação de preferência de estudantes superdotadas
As noções de masculino e feminino: concepções ideológicas e papéis de gênero.
Mulheres mulas do tráfico: estudo sobre a lei 11.343/06 sob uma perspectiva de gênero.
Diversidade, imigração e desenvolvimento.
Concepções de acadêmicos e acadêmicas de licenciatura em ciências biológicas a respeito da temática de diversidade de gênero e sexualidade.

tarefas, com cada tarefa contendo de 3 a 6 passo. As tarefas 1-3 buscam fazer o usuário explorar as 3 perguntas principais (teses, artigos e revistas), bem como as 3 operações (busca, filtragem e clarificação). As tarefas 4-6 exigem que o usuário utilize as operações em conjunto como, por exemplo, buscar por teses de um tema de interesse (busca), perguntar pelo orientador de algum resultado (clarificação), e filtrar os resultados da busca pelo nome do orientador (filtrar) para encontrar todas as teses orientadas por ele nesse tema.

No questionário de avaliação, foram desenvolvidas questões de fácil entendimento, avaliando os aspectos que consideramos mais relevantes do *chatbot*, propondo a atribuição de graus variando de 1 a 5 para cada aspecto, sendo esses:

1. Utilização do *chatbot* (difícil a fácil)
2. Consistência das respostas (sem consistência a consistente)
3. Utilidade do *chatbot* (inútil a útil)
4. *Tempo de resposta* do *chatbot* (lento a rápido)
5. *Ordenação* dos resultados da busca quanto a similaridade com o termo (mal ordenados a bem ordenados)
6. Listagem de resultados semanticamente similares em outras linguagens (mal listados a bem listados)
7. Funcionalidade da operação de filtragem (não-funcional a funcional)
8. Funcionalidade da operação de clarificação (não-funcional a funcional)
9. *Utilidade* de interfaces conversacionais para obtenção de informação da plataforma Sucupira, dado a experiência com o *SucupiraBot* (inútil a útil)

O questionário foi aplicado após a realização das tarefas, com objetivo de tentar quantificar a experiência dos usuários com o *chatbot*.

Os resultados estão resumidos na Figura 7.5, que indica a porcentagem de participantes que deu cada nota (de 1 a 5 – eixo X) para cada item (eixo Y). A nota de grau 3 pode ser vista como opinião neutra. Na média, o bot obteve mais avaliações positivas que negativas. As questões de utilização e consistência do bot receberam votações balanceadas entre graus 3 a 5. com maior concentração no grau 4, indicando um posicionamento positivo quanto a facilidade de se utilizar e consistência nas resposta, embora não seja majoritariamente definido como fácil ou consistente. As questões sobre

avaliação da utilidade do *chatbot* e percepção do tempo de resposta foram majoritariamente positivas, e é interessante destacar múltiplos comentários dos participantes elogiando a rápida resposta do *chatbot*.

As questões 5 e 6, sobre ordenação das respostas do *chatbot*, tem como objetivo avaliar o modelo de vetores de palavras, e e obteve resultados positivos. Já a questão 6 foi mais polarizada, com concentração de votos nos graus 3 e 5.

As questões 7 e 8 avaliam o uso das funcionalidades de filtragem e clarificação. Os resultados também tenderam a ser mais positivos, indicando ao menos o funcionamento das operações nas tarefas propostas com algum ajuste. Um participante relatou não conseguir utilizar bem a clarificação, resultando na porcentagem do grau 1.

Por fim, a questão 9 tem como objetivo mensurar a percepção dos usuários quanto ao uso de interfaces conversacionais na recuperação de informação do Sucupira. A maioria absoluta votou na utilidade desse meio, com muitos comentários elogiando a iniciativa e empolgados com possíveis melhorias que virão no futuro.

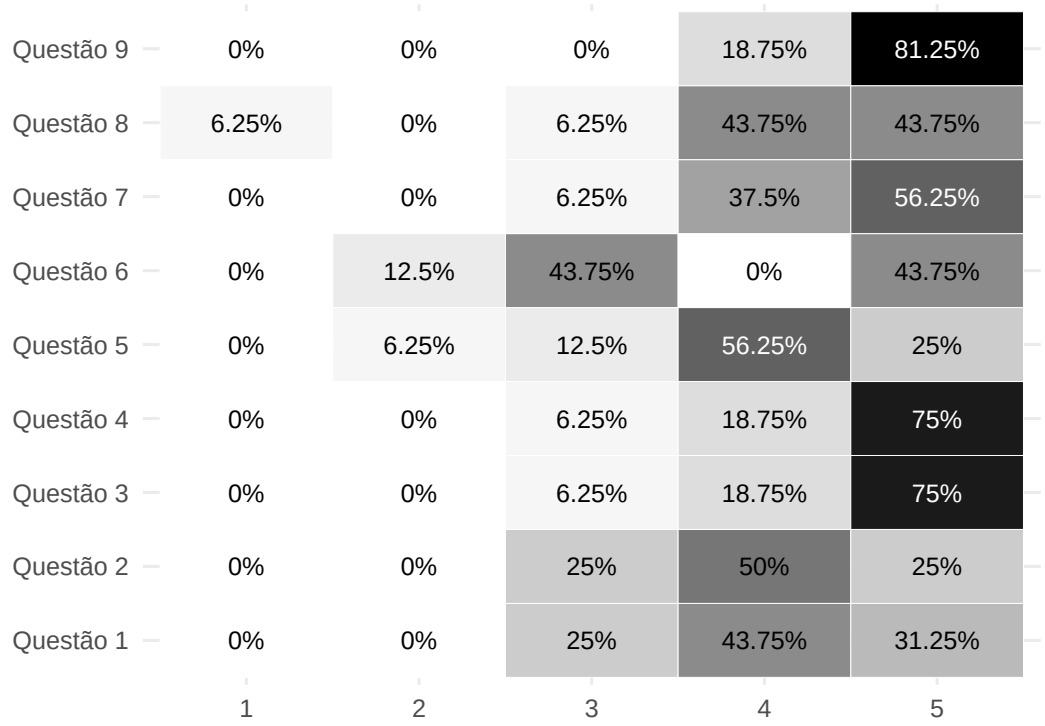


Figura 7.5: Sumário das respostas ao questionário.

7.5 Discussão

Este capítulo procurou demonstrar o uso dos modelos treinados em um caso real de aplicação, fornecendo assim mais formas de avaliação. A interação

com usuários foi positiva, abrindo vários caminhos para inovações no sistema e nas tarefas.

Os tipos de questões que o *QA* responde são de uma categoria considerada "simples", com no máximo 3 tipos de entidades diferentes, sendo duas de dados e uma de busca. Ainda não há uma modelagem explícita dos relacionamentos entre entidades, permitindo perguntas como: "Quero teses que falam sobre *cannabis* e artigos sobre as eleições de 2018". Em suma, a área de *QA* possui uma gama de particularidades e de tarefas[76] que estão fora do escopo para uma simples demonstração de aplicação.

O *IQA* também possui muitas limitações quanto a questão de *coreference* [74][75], sendo este um campo desafiador da área de NLP. É necessário abordar técnicas mais apropriadas a esse domínio.

O *dataset* de treinamento de entidades e intenções ainda está em desenvolvimento, tendo-se o objetivo de aumentar o número de *templates* para diversificá-lo ainda mais. Por fim, é de interesse em trabalhos futuros abordar essas e outras limitações, bem como tarefas mais difíceis.

8
Considerações Finais

Esse trabalho apresentou o processo de construção do *corpus* acadêmico de teses e dissertações, partindo inicialmente do levantamento bibliográfico de corpora propostos na língua portuguesa, sendo de conhecimento geral ou acadêmico. Uma metodologia de coleta e pré-processamento dos dados foi proposta, seguido pelo uso do Grobid para extração do conteúdo textual e limpeza dos dados e posterior analise. Modelos com vetores de palavras estáticos e modelos contextuais, além de modelos clássicos como tf-idf, foram treinados e avaliados, sendo que o modelo *transformer* BERTAcademico obteve os melhores resultados, mostrando que o conhecimento contextual ajuda nas tarefas propostas.

Também foi feita a exploração do corpus por meio da modelagem de tópicos, onde se extraiu 10 tópicos coerentes a partir do modelo LDA. Essa exploração foi relevante ao ser a primeira analise inicial da riqueza semântica escondida no corpus. Em seguida, foi feita uma redução de dimensionalidade por meio do UMAP, e foi possível visualizar os vetores dos resumos das áreas de conhecimento em um espaço de duas dimensões.

Foi feita também uma aplicação que faz uso dos modelos treinados, o QA Interativo SucupiraBot, no qual tarefas de classificação e NER são parte do sistema proposto, e uma boa performance foi obtida com os modelos. Uma avaliação por meio de questionário foi realizada, obtendo-se um bom *feedback* dos usuários.

Vários trabalhos futuros podem ser explorados. Por exemplo, o *corpus* pode ser dividido em corporas correspondendo as grande áreas de conhecimento, ou até mesmo as áreas de conhecimento. A partir dessas categorizações, modelos de linguagem para cada domínio pode ser treinado, sendo possível assim explorar visualizações mais refinadas, além dos tópicos poderem ser mais focados nas áreas de interesse. Tarefas que exigem um conhecimento mais especializado dos modelos podem usufruir dos vetores de palavras ou de modelos pré-treinados no *corpus*(BERTAcadêmico). Por fim, é importante ressaltar que existem possibilidades além das citadas anteriormente tanto em aspecto de extração de conhecimento do *corpus* quanto em construção de modelos que fazem uso desse conhecimento.

Referências bibliográficas

[1] PEREIRA, I.; SOUSA, J.; COSTA, P. B.; BARBOSA, S. D. ; COLCHER, S.. **Sucupirabot: An interactive question-answering system for the sucupira platform**. In: PROCEEDINGS OF THE BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB, p. 277–280, 2020.

[2] RUMELHART, D. E.; HINTON, G. E. ; WILLIAMS, R. J.. **Learning representations by back-propagating errors**. nature, 323(6088):533–536, 1986.

[4] SANG, E. F.; DE MEULDER, F.. **Introduction to the conll-2003 shared task: Language-independent named entity recognition**. arXiv preprint cs/0306050, 2003.

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

[6] NEVES, M.; YEPES, A. J. ; NÉVÉOL, A.. **The scielo corpus: a parallel corpus of scientific publications for biomedicine**. In: PROCEEDINGS OF THE TENTH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC'16), p. 2942–2948, 2016.

[7] AZIZ, W.; SPECIA, L.. **Fully automatic compilation of a portuguese-english parallel corpus for statistical machine translation**. In: IN STIL 2011. Citeseer, 2011.

[8] SANTOS, H.; WOLOSZYN, V. ; VIEIRA, R.. **Blogset-br: a brazilian portuguese blog corpus**. In: PROCEEDINGS OF THE ELEVENTH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC 2018), 2018.

[9] WAGNER FILHO, J. A.; WILKENS, R.; IDIART, M. ; VILLAVICENCIO, A.. **The brwac corpus: A new open resource for brazilian portuguese**. In: PROCEEDINGS OF THE ELEVENTH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC 2018), 2018.

[10] SUÁREZ, P. J. O.; SAGOT, B. ; ROMARY, L.. **Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures**. In: 7TH WORKSHOP ON THE CHALLENGES IN THE MANAGEMENT OF LARGE CORPORA (CMLC-7). Leibniz-Institut für Deutsche Sprache, 2019.

[11] RODRIGUES, J.; BRANCO, A.; NEALE, S. ; SILVA, J.. **Lx-dsemvectors: Distributional semantics models for portuguese**. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL PROCESSING OF THE PORTUGUESE LANGUAGE, p. 259–270. Springer, 2016.

[12] PARDO, T. A. S.; NUNES, M. D. G. V. ; RINO, L. H. M.. **Dizer: An automatic discourse analyzer for brazilian portuguese**. In: BRAZILIAN SYMPOSIUM ON ARTIFICIAL INTELLIGENCE, p. 224–234. Springer, 2004.

[13] CONSOLI, B.; SANTOS, J.; GOMES, D.; CORDEIRO, F.; VIEIRA, R. ; MOREIRA, V.. **Embeddings for named entity recognition in geoscience portuguese literature**. In: PROCEEDINGS OF THE 12TH LANGUAGE RESOURCES AND EVALUATION CONFERENCE, p. 4625–4630, 2020.

[14] RODRIGUES, J.; BRANCO, A.. **Finely tuned, 2 billion token based word embeddings for portuguese**. In: PROCEEDINGS OF THE ELEVENTH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION (LREC 2018), 2018.

[15] KOHLMEIER, S.; LO, K.; WANG, L. ; YANG, Y.. **Covid-19 open research dataset challenge (cord-19)**. Zenodo, p. e, 2020.

[16] HAYKIN, S.. **Neural Networks and Learning Machines**. Pearson Education India, 2010.

[17] MURPHY, K. P.. **Machine Learning: A Probabilistic Perspective**. MIT Press, 2012.

[18] GOODFELLOW, I.; BENGIO, Y. ; COURVILLE, A.. **Deep learning**. MIT press, 2016.

[19] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł. ; POLOSUKHIN, I.. **Attention is all you need**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 5998–6008, 2017.

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

[21] LIU, Y.; OTT, M.; GOYAL, N.; DU, J.; JOSHI, M.; CHEN, D.; LEVY, O.; LEWIS, M.; ZETTLEMOYER, L. ; STOYANOV, V.. **Roberta: A robustly optimized bert pretraining approach**. arXiv preprint arXiv:1907.11692, 2019.

[22] GAO, J.; GALLEY, M. ; LI, L.. **Neural Approaches to Conversational AI**. Foundations and Trends® in Information Retrieval, 13(2-3):127–298, sep 2018.

[23] AUER, S.; BIZER, C.; KOBILAROV, G.; LEHMANN, J.; CYGANIAK, R. ; IVES, Z.. **Dbpedia: A nucleus for a web of open data**. In: THE SEMANTIC WEB, p. 722–735. Springer, 2007.

[24] VRANDEČIĆ, D.; KRÖTZSCH, M.. **Wikidata: a free collaborative knowledgebase**. Communications of the ACM, 57(10):78–85, 2014.

[25] SUCHANEK, F. M.; KASNECI, G. ; WEIKUM, G.. **Yago: a core of semantic knowledge**. In: PROCEEDINGS OF THE 16TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, p. 697–706, 2007.

[26] LOPEZ, P.. **Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications**. In: INTERNATIONAL CONFERENCE ON THEORY AND PRACTICE OF DIGITAL LIBRARIES, p. 473–474. Springer, 2009.

[28] MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S. ; DEAN, J.. **Distributed representations of words and phrases and their compositionality**. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, p. 3111–3119, 2013.

[29] PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K. ; ZETTLEMOYER, L.. **Deep contextualized word representations**. arXiv preprint arXiv:1802.05365, 2018.

[30] BELTAGY, I.; LO, K. ; COHAN, A.. **Scibert: A pretrained language model for scientific text**. arXiv preprint arXiv:1903.10676, 2019.

[31] LEE, J.; YOON, W.; KIM, S.; KIM, D.; KIM, S.; SO, C. H. ; KANG, J.. **Biobert: a pre-trained biomedical language representation**

model for biomedical text mining. Bioinformatics, 36(4):1234–1240, 2020.

[32] ELMAN, J. L.. **Finding structure in time**. Cognitive science, 14(2):179–211, 1990.

[33] BENGIO, Y.; SIMARD, P. ; FRASCONI, P.. **Learning long-term dependencies with gradient descent is difficult**. IEEE transactions on neural networks, 5(2):157–166, 1994.

[34] PHUNG, V. H.; RHEE, E. J.. **A deep learning approach for classification of cloud image patches on small datasets**. Journal of information and communication convergence engineering, 16(3):173–178, 2018.

[35] **Understanding lstm networks**.

[36] **Understanding attention mechanism**.

[37] ALAMMAR, J.. **The illustrated transformer**.

[38] MIKOLOV, T.; LE, Q. V. ; SUTSKEVER, I.. **Exploiting similarities among languages for machine translation**. arXiv preprint arXiv:1309.4168, 2013.

[40] BOJANOWSKI, P.; GRAVE, E.; JOULIN, A. ; MIKOLOV, T.. **Enriching word vectors with subword information**. Transactions of the Association for Computational Linguistics, 5:135–146, 2017.

[41] SAINBURG, T.; MCINNES, L. ; GENTNER, T. Q.. **Parametric umap embeddings for representation and semi-supervised learning**. arXiv preprint arXiv:2009.12981, 2020.

[42] MIKOLOV, T.; CHEN, K.; CORRADO, G. ; DEAN, J.. **Efficient estimation of word representations in vector space**. arXiv preprint arXiv:1301.3781, 2013.

[43] SCHUSTER, M.; NAKAJIMA, K.. **Japanese and korean voice search**. In: 2012 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), p. 5149–5152. IEEE, 2012.

[44] SENNRICH, R.; HADDOW, B. ; BIRCH, A.. **Neural machine translation of rare words with subword units**. arXiv preprint arXiv:1508.07909, 2015.

[45] SALTON, G.; FOX, E. A. ; WU, H.. **Extended boolean information retrieval**. Communications of the ACM, 26(11):1022–1036, 1983.

[46] EL-KASSAS, W. S.; SALAMA, C. R.; RAFEA, A. A. ; MOHAMED, H. K.. **Automatic text summarization: A comprehensive survey**. Expert Systems with Applications, 165:113679, 2021.

[49] KHAN, S.; LIU, X.; SHAKIL, K. A. ; ALAM, M.. **A survey on scholarly data: From big data perspective**. Information Processing & Management, 53(4):923–944, 2017.

[50] NASAR, Z.; JAFFRY, S. W. ; MALIK, M. K.. **Information extraction from scientific articles: a survey**. Scientometrics, 117(3):1931–1990, 2018.

[51] HUANG, Z.; XU, W. ; YU, K.. **Bidirectional lstm-crf models for sequence tagging**. arXiv preprint arXiv:1508.01991, 2015.

[52] RADFORD, A.; NARASIMHAN, K.; SALIMANS, T. ; SUTSKEVER, I.. **Improving language understanding by generative pre-training**. 2018.

[53] SOUZA, F.; NOGUEIRA, R. ; LOTUFO, R.. **BERTimbau: pretrained BERT models for Brazilian Portuguese**. In: 9TH BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS, BRACIS, RIO GRANDE DO SUL, BRAZIL, OCTOBER 20-23 (TO APPEAR), 2020.

[54] TALBI, E.-G.. **Optimization of deep neural networks: a survey and unified taxonomy**. 2020.

[55] LIU, P.; WANG, X.; XIANG, C. ; MENG, W.. **A survey of text data augmentation**. In: 2020 INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATION AND NETWORK SECURITY (CCNS), p. 191–195. IEEE, 2020.

[56] BLEI, D. M.; NG, A. Y. ; JORDAN, M. I.. **Latent dirichlet allocation**. the Journal of machine Learning research, 3:993–1022, 2003.

[57] MCLACHLAN, G. J.; BASFORD, K. E.. **Mixture models: Inference and applications to clustering**, volumen 38. M. Dekker New York, 1988.

[58] GELFAND, A. E.. **Gibbs sampling**. Journal of the American statistical Association, 95(452):1300–1304, 2000.

[59] LOPER, E.; BIRD, S.. **Nltk: The natural language toolkit**. arXiv preprint cs/0205028, 2002.

[60] MCINNES, L.; HEALY, J. ; MELVILLE, J.. **Umap: Uniform manifold approximation and projection for dimension reduction**. arXiv preprint arXiv:1802.03426, 2018.

[62] TEH, Y. W.; JORDAN, M. I.; BEAL, M. J. ; BLEI, D. M.. **Hierarchical dirichlet processes**. Journal of the american statistical association, 101(476):1566–1581, 2006.

[63] LI, W.; MCCALLUM, A.. **Pachinko allocation: Dag-structured mixture models of topic correlations**. In: PROCEEDINGS OF THE 23RD INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 577–584, 2006.

[64] BLEI, D. M.; GRIFFITHS, T. L.; JORDAN, M. I.; TENENBAUM, J. B. ; OTHERS. **Hierarchical topic models and the nested chinese restaurant process.** In: NIPS, volumen 16, 2003.

[65] LIU, X.; SONG, Y.; LIU, S. ; WANG, H.. **Automatic taxonomy construction from keywords**. In: PROCEEDINGS OF THE 18TH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, p. 1433–1441, 2012.

[66] AL-ASWADI, F. N.; CHAN, H. Y. ; GAN, K. H.. **Automatic ontology construction from text: a review from shallow to deep learning trend**. Artificial Intelligence Review, 53(6):3901–3928, 2020.

[67] BIENZ, T.; COHN, R. ; ADOBE SYSTEMS (MOUNTAIN VIEW, C.. **Portable document format reference manual**. Citeseer, 1993.

[68] PERRY, T. S.. **'postscript'prints anything: a case history**. IEEE Spectrum, 25(5):42–46, 1988.

[69] BLEI, D. M.. **Probabilistic topic models**. Communications of the ACM, 55(4):77–84, 2012.

[71] HARTMANN, A.-K.; MARX, E. ; SORU, T.. **Generating a large dataset for neural question answering over the dbpedia knowledge base**. In: WORKSHOP ON LINKED DATA MANAGEMENT, CO-LOCATED WITH THE W3C WEBBR, volumen 2018, 2018.

[72] 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.

[73] SOARES, F.; YAMASHITA, G. H. ; ANZANELLO, M. J.. **A parallel corpus of theses and dissertations abstracts**. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL PROCESSING OF THE PORTUGUESE LANGUAGE, p. 345–352. Springer, 2018.

[74] ELANGO, P.. **Coreference resolution: A survey**. University of Wisconsin, Madison, WI, 2005.

[75] STYLIANOU, N.; VLAHAVAS, I.. **A neural entity coreference resolution review**. Expert Systems with Applications, 168:114466, 2021.

[76] SOARES, M. A. C.; PARREIRAS, F. S.. **A literature review on question answering techniques, paradigms and systems**. Journal of King Saud University-Computer and Information Sciences, 32(6):635–646, 2020.

[77] 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), p. 1532–1543, 2014.

[79] HU, D.. **An introductory survey on attention mechanisms in nlp problems**. In: PROCEEDINGS OF SAI INTELLIGENT SYSTEMS CONFERENCE, p. 432–448. Springer, 2019.

[80] NASAR, Z.; JAFFRY, S. W. ; MALIK, M. K.. **Named entity recognition and relation extraction: State-of-the-art**. ACM Computing Surveys (CSUR), 54(1):1–39, 2021.

A
Apêndice

O Apêndice contem os resultados da classificação das Áreas de conhecimento, bem como os 10 tópicos descobertos com mais termos.

Tabela A.1: F1 por Área de Conhecimento

Área Conhecimento	tfidf	word2vec	fasttext	BERTAcadêmico	Bertimbau
ADMINISTRAÇÃO	0.69	0.68	0.70	**0.72**	0.66
AGRONOMIA	0.74	0.73	0.73	**0.75**	0.71
ANTROPOLOGIA	0.50	0.49	0.49	**0.65**	0.49
ARQUEOLOGIA	0.42	0.40	0.42	**0.51**	0.40
ARQUITETURA E URBANISMO	0.62	0.60	0.61	**0.65**	0.56
ARTES	0.72	0.72	0.72	**0.73**	0.68
ASTRONOMIA	0.51	0.50	0.51	0.51	0.43
BIOFÍSICA	0.17	0.15	0.16	**0.25**	0.13
BIOLOGIA GERAL	0.31	0.30	0.32	**0.34**	0.25
BIOQUÍMICA	0.37	0.37	0.36	**0.42**	0.36
BIOTECNOLOGIA	0.23	0.21	0.22	**0.27**	0.20
BOTÂNICA	0.57	0.56	0.57	**0.58**	0.51
CIÊNCIA DA COMPUTAÇÃO	0.70	0.68	0.70	**0.71**	0.68
CIÊNCIA DA INFORMAÇÃO	0.69	0.69	0.68	**0.70**	0.65
CIÊNCIA E TECNOLOGIA DE ALIMENTOS	0.63	0.62	0.64	**0.65**	0.62
CIÊNCIA POLÍTICA	0.52	0.51	0.51	**0.53**	0.50
CIÊNCIAS AMBIENTAIS	0.31	0.30	0.31	**0.33**	0.30
COMUNICAÇÃO	0.65	0.64	0.64	**0.66**	0.55
DEMOGRAFIA	0.41	0.40	0.41	**0.42**	0.40
DESENHO INDUSTRIAL	0.63	0.60	0.63	**0.64**	0.59
DIREITO	0.85	0.83	0.85	**0.86**	0.82
ECOLOGIA	0.51	0.51	0.51	**0.52**	0.45
ECONOMIA	0.67	0.66	0.67	**0.68**	0.62

Tabela A.1: F1 por Área de Conhecimento

Área Conhecimento	tfidf	word2vec	fasttext	BERTAcadêmico	Bertimbau
ECONOMIA DOMÉSTICA	0.00	0.00	0.00	0.00	0.00
EDUCAÇÃO	0.71	0.71	0.71	0.71	0.69
EDUCAÇÃO FÍSICA	0.59	0.58	0.59	**0.62**	0.56
ENFERMAGEM	0.69	0.66	0.69	**0.71**	0.68
ENGENHARIA AEROESPACIAL	0.43	0.42	0.42	**0.43**	0.40
ENGENHARIA AGRÍCOLA	0.37	0.37	0.37	**0.38**	0.35
ENGENHARIA BIOMÉDICA	0.23	0.23	0.22	**0.31**	0.21
ENGENHARIA CIVIL	0.62	0.61	0.62	**0.63**	0.60
ENGENHARIA DE MATERIAIS E METALÚRGICA	0.49	0.49	0.50	**0.51**	0.44
ENGENHARIA DE MINAS	0.21	0.20	0.22	**0.25**	0.20
ENGENHARIA DE PRODUÇÃO	0.39	0.38	0.39	**0.41**	0.37
ENGENHARIA DE TRANSPORTES	0.34	0.32	0.34	**0.36**	0.32
ENGENHARIA ELÉTRICA	0.67	0.66	0.67	**0.68**	0.66
ENGENHARIA MECÂNICA	0.56	0.56	0.56	**0.58**	0.55
ENGENHARIA NAVAL E OCEÂNICA	0.36	0.36	0.36	**0.38**	0.35
ENGENHARIA NUCLEAR	0.52	0.51	0.52	**0.52**	0.50
ENGENHARIA QUÍMICA	0.50	0.50	0.50	**0.51**	0.48
ENGENHARIA SANITÁRIA	0.33	0.31	0.34	**0.37**	0.31
ENSINO	0.59	0.58	0.58	**0.60**	0.57
FARMACOLOGIA	0.35	0.34	0.35	**0.36**	0.33
FARMÁCIA	0.50	0.50	0.50	**0.51**	0.48
FILOSOFIA	0.76	0.75	0.76	**0.78**	0.74

Tabela A.1: F1 por Área de Conhecimento

Área Conhecimento	tfidf	word2vec	fasttext	BERTAcadêmico	Bertimbau
FISIOLOGIA	0.41	0.39	0.**42**	0.41	0.34
FISIOTERAPIA E TERAPIA OCUPACIONAL	0.46	0.46	0.46	0.46	0.40
FONOAUDIOLOGIA	0.50	0.49	0.50	**0.51**	0.47
FÍSICA	0.73	0.71	0.72	**0.74**	0.68
GENÉTICA	0.37	0.36	0.35	**0.37**	0.33
GEOCIÊNCIAS	0.70	0.70	0.70	**0.70**	0.65
GEOGRAFIA	0.60	0.59	0.61	**0.62**	0.60
HISTÓRIA	0.67	0.66	0.67	**0.69**	0.64
IMUNOLOGIA	0.31	0.30	0.32	**0.34**	0.30
INTERDISCIPLINAR	0.24	0.23	0.23	**0.26**	0.21
LETRAS	0.75	0.75	0.75	**0.76**	0.72
LINGÜÍSTICA	0.46	0.46	0.46	**0.47**	0.41
MATEMÁTICA	0.86	0.85	0.85	**0.88**	0.83
MATERIAIS	0.24	0.23	0.24	**0.27**	0.22
MEDICINA	0.67	0.64	0.67	**0.68**	0.62
MEDICINA VETERINÁRIA	0.68	0.66	0.67	**0.70**	0.66
MICROBIOLOGIA	0.37	0.37	0.37	**0.39**	0.32
MORFOLOGIA	0.22	0.21	0.22	**0.24**	0.20
MUSEOLOGIA	0.48	0.46	0.48	**0.48**	0.33
NUTRIÇÃO	0.37	0.36	0.37	**0.38**	0.33
OCEANOGRAFIA	0.35	0.35	0.34	**0.36**	0.32
ODONTOLOGIA	0.87	0.85	0.86	**0.88**	0.84
PARASITOLOGIA	0.24	0.22	0.23	**0.25**	0.21
PLANEJAMENTO URBANO E REGIONAL	0.23	0.22	0.22	**0.25**	0.21
PROBABILIDADE E ESTATÍSTICA	0.64	0.63	0.65	**0.66**	0.64
PSICOLOGIA	0.66	0.65	0.67	**0.68**	0.66
QUÍMICA	0.66	0.66	0.67	**0.67**	0.64
RECURSOS FLORESTAIS E ENGENHARIA FLORESTAL	0.59	0.58	0.59	**0.60**	0.57

Tabela A.1: F1 por Área de Conhecimento

Área Conhecimento	tfidf	word2vec	fasttext	BERTAcadêmico	Bertimbau
RECURSOS PESQUEIROS E ENGENHARIA DE PESCA	0.49	0.47	0.49	**0.51**	0.46
SAÚDE COLETIVA	0.47	0.47	0.47	**0.48**	0.42
SERVIÇO SOCIAL	0.59	0.59	0.60	0.60	0.57
SOCIOLOGIA	0.34	0.32	0.34	**0.36**	0.32
TEOLOGIA	0.68	0.67	0.69	**0.70**	0.66
TURISMO	0.58	0.57	0.59	**0.60**	0.57
ZOOLOGIA	0.51	0.50	0.51	**0.55**	0.50
ZOOTECNIA	0.56	0.55	0.56	0.56	0.54

Tabela A.2: Tópicos(Top 10) completos

Tópico	**Termos**
1	sociedade, sentido, discurso, social, sujeito, modo, espaco, perspectiva, discursos, sociais, praticas, compreensao, sujeitos, construcao, pensar, campo, homem, conceito, processo, realidade, lugar, identidade, experiencia, etica, ideia, fenomeno, corpo, saber, sentidos, subjetividade
2	educacao, escola, formacao, professores, ensino, escolas, escolar, docentes, formacao_professores, formacao_continuada, curriculo, docente, professoras, educacional, educacao_fisica, professor, educacao_infantil, pedagogia, curso, cursos, alunos, pratica, sujeitos, saberes, pedagogico, pratica_pedagogica, profissional, formacao_inicial, praticas_pedagogicas, educacionais
3	saude, profissionais, cuidado, familia, enfermagem, enfermeiros, adolescentes, pessoas, profissionais_saude, familiares, mulheres, atendimento, assistencia, cuidados, atencao, crianca, equipe, violencia, percepcao, coleta_dados, participantes, idosos, medicos, saude_mental, filhos, usuarios, enfermeiro, entrevistas, mental, entrevista
4	desenvolvimento, projeto, metodologia, design, estudos, processo, aplicacao, projetos, proposta, capitulo, planejamento, revisao, definicao, criterios, elaboracao, aspectos, procedimentos, tecnicas, conhecimento, metodos, identificacao, base, pesquisas, construcao, conceitos, informacoes, etapas, metodo, abordagem, metodologias

5	politicas, politica, politicas_publicas, cidade, social, estado, trabalhadores, territorio, participacao, espaco, municipio, cidades, sociais, luta, desenvolvimento, turismo, espacos, movimentos_sociais, municipios, acoes, processo, rural, decada, governo, moradores, politica_publica, programa, capital, atores, organizacao
6	obra, arte, romance, narrativa, personagens, literatura, narrativas, cinema, imagens, obras, discurso, teatro, textos, texto, poetica, poesia, midia, leitura, personagem, jornalismo, escrita, imagem, escritor, literaria, contos, romances, estetica, linguagem, livro, artista
7	animais, ratos, induzida, tratamento, efeitos, camundongos, estresse_oxidativo, administracao, dose, ratos_wistar, aumento, animais_tratados, induzido, insulina, tratados, animais_submetidos, ratas, receptores, controle, oxido_nitrico, niveis, injecao, resposta, estresse, doses, inducao, figado, inflamacao, via, modelo_experimental
8	escoamento, modelo, simulacao, metodo_elementos, elementos_finitos, comportamento, velocidade, fluido, obtidos, tensoes, medicacao, pressao, ensaios, temperatura, vigas, parametros, calor, modelos, simulacoes, geometria, equipamento, finitos, fluxo, carregamento, dimensionamento, metodo, transferencia_calor, tensao, equacoes, escoamentos
9	direito, direitos, lei, justica, juridica, juridico, judiciario, constitucional, constituicao_federal, direitos_humanos, direitos_fundamentais, ordenamento_juridico, estado, principio, normas, estado_democratico, instituto, protecao, penal, legislacao, tutela, principios, constituicao, contrato, doutrina, direito_fundamental, supremo_tribunal, dignidade_pessoa, civil, norma
10	software, redes, sistemas, arquitetura, aplicacoes, internet, rede, usuario, dados, usuarios, comunicacao, web, ambiente, ferramenta, computacao, ferramentas, plataforma, informacao, dispositivos, informacoes, implementacao, tecnologia, objetos, ambientes, tecnologias, computadores, digital, requisitos, processamento, modelo