# Introdução

Algoritmos baseados em aprendizado de máquina são, por definição, dependentes de um conjunto de dados para a realização do seu treinamento. Não é diferente com os algoritmos de processamento de linguagem natural (PLN). No entanto, há uma grande dificuldade para o desenvolvimento de modelos de PLN pela indústria do petróleo no Brasil. Há uma carência de bases de documentos de referência que possam servir como conjunto de dados para a realização desses treinamentos. Portanto, o objetivo desse trabalho foi criar um corpus público formado por teses e dissertações em português, no domínio de óleo e gás que sirva de referência para os grupos de pesquisas em inteligência artificial e empresas relacionadas à indústria do petróleo.

Para a criação desse corpus, foram utilizados documentos disponíveis na Biblioteca Digital de Teses e Dissertações (BDTD) do Instituto Brasileiro de Informação em Ciência e Tecnologia (IBICT). No entanto, como a BDTD possui quase 600 mil documentos de todas as áreas do conhecimento, foi necessário identificar se um documento era relevante para a indústria de óleo e gás antes de extraí-lo da base. Todo esse processo foi feito automaticamente utilizando um modelo treinado para classificar o resumo de um documento e um robô para processar centenas de milhares de teses e dissertações presentes na BDTD.

O corpus final é composto por 4.302 documentos, contendo cerca de 6 milhões de sentenças e 74,4 milhões de tokens. Esse novo corpus servirá como referência para o desenvolvimento dos futuros trabalhos de processamento de linguagem na indústria do petróleo.

# Corpus

Todos os algoritmos de aprendizado de máquina dependem de dados para o seu treinamento. No caso dos algoritmos de processamento de linguagem, os dados de treinamento são conjuntos de textos. Um conjunto de textos, de forma geral, é chamado de corpus, e seu plural de corpora. Dependendo do objetivo, os corpora podem ter diversas características, por exemplo eles podem ser monolíngues ou multilíngues, podem ser transcrições da linguagem falada no dia a dia ou serem compostos de textos técnicos de uma área do conhecimento específica. A escolha ou criação de um corpus depende do uso e tarefas que se pretende.

Segundo *The Oxford Companion to the English*, corpus é:

“Um corpo de textos, enunciados ou outros espécimes considerados mais ou menos representativos da linguagem, e usualmente armazenados em uma base de dados eletrônica. Atualmente, corpora computacionais podem armazenar muitos milhões de palavras corridas, no qual as suas características podem ser analisadas por meio de marcações (adição de marcas de identificação e classificação nas palavras e outras formações) e o uso de programas de concordância. ” (McArthur, 1992, tradução nossa)[[1]](#footnote-1)

Os corpora podem ser classificados de acordo com o seu conteúdo, seus metadados, presença de multimídia e sua relação com outros corpora. De acordo com a Lexical Computing (2019), podemos classificar os corpora da seguinte maneira:

* Corpus monolíngue – são aqueles compostos por textos em apenas um idioma. São os tipos de corpora mais comum.
* Corpus paralelo – é composto por dois corpora monolíngue em idiomas diferentes, sendo que um é a tradução do outro. Os textos devem estar alinhados permitindo observar como as correspondências dos segmentos, em geral palavras e sentenças, aparecem nos dois idiomas.
* Corpus multilíngue – similar ao corpus paralelo, podendo ser composto por corpora em várias línguas. Um corpus multilíngue com apenas dois idiomas é idêntico a um corpus paralelo.
* Corpus comparável – é composto por dois ou mais corpus monolíngue cujos textos são relacionados ao mesmo tópico, no entanto não são traduções um do outro. Um exemplo são corpora composto por verbetes da Wikipédia.
* Corpus de aprendiz – é formado por textos produzidos por pessoas que estão aprendendo uma língua estrangeira. É utilizado para identificar os principais erros e dificuldades no aprendizado de uma nova língua.
* Corpus diacrônicos – contém documentos de diferentes períodos de tempo. É usado para o estudo do desenvolvimento ou mudança da linguagem.

Complementando essas categorias, um corpus também pode ser classificado como:

* Especializado – quando é formado por documentos exclusivos de uma área do conhecimento ou de um domínio específico.
* Multimídia – corpora formados por textos e enriquecidos com áudios, imagens e outros conteúdos multimídia.

Para as atividades de processamento de linguagem natural mais comuns, há algumas características desejáveis em um corpus. Uma delas é a profundidade, ou seja, é importante que a quantidade de palavras total do corpus bem como o tamanho do vocabulário (número de palavras únicas) sejam relativamente grandes. Um corpus muito “raso” pode não conseguir captar a diversidade linguística presente nos textos. Outra característica importante é a data de criação dos textos. Exceto para corpora diacrônicos, em geral, é desejável textos recentes. Os metadados que acompanham os textos podem trazer informações valiosas e podem auxiliar em diversas tarefas de aprendizado supervisionado. Para muitas aplicações também é necessária uma pluralidade de gêneros e tipos de texto, evitando que haja um viés proveniente de documentos de um mesmo tipo ou fonte. Finalmente, para algumas aplicações é útil que o corpus já esteja pré-processado e limpo. (PADMANABHAN, 2019)

Alguns dos elementos básicos para o processamento automático da linguagem podem ser obtidos através de corpora bem curados e montados. Exemplos desses elementos são listas de palavras, classes gramaticais e modelos mais complexos de representação de palavras usando espaços vetoriais.

As palavras são os blocos básicos que constituem um corpus. Obter uma lista do vocabulário presente em um corpus, bem como a frequência em que as palavras aparecem pode ser útil em vários contextos. Por exemplo, conhecer quais são as *stopwords*, ou seja, palavras que trazem pouca informação e acrescentam pouco na predição ou recuperação da informação (Makrehchi e Kamel, 2008), é útil para diversas tarefas de limpeza e pré-processamento de textos. No entanto, além de conhecer as palavras, também pode ser interessante conhecer a estrutura gramatical das sentenças e textos. Por isso, muitos corpora são manualmente anotados com as estruturas gramaticais (*Part-of-speech tagging*) e com as relações entre essas estruturas (*Treebanks*). (PADMANABHAN, 2019)

Finalmente, pode-se utilizar um corpus para se criar uma representação matemática das palavras de forma a capturar algumas relações semânticas presente nos textos. Essas representações matemáticas, em geral vetores, podem servir de entrada para diversos modelos de aprendizado de máquina. Em 2013, Mikolov et al. apresentaram o modelo de vetorização de palavras utilizando redes neurais mais utilizado atualmente, o Word2Vec. Nesse modelo, todas as palavras são representadas como vetores e os valores desses vetores é baseado nas palavras vizinhas. Ou seja, as palavras deixam de ser apenas representações gráficas e passam a capturar parte do contexto das sentenças onde estão inseridas. Desta forma, o conjunto de textos utilizados para treinar esses vetores é fundamental para a qualidade final dessa representação.

Mais recentemente, arquiteturas de *deep learning* mais complexas estão sendo testadas para capturar as relações de dependências entre as palavras em documentos escritos em linguagem natural. Atualmente, o modelo BERT (*Bidirectional Encoder Representations from Transformers*) (Devlin et al.,2019) tenta capturar as relações semânticas do texto onde ele é treinado e, atualmente, tem alcançado os melhores resultados em diversas tarefas de processamento de linguagem natural. Assim como os modelos de vetorização de palavras, os modelos de *deep learning* são altamente dependentes da qualidade do corpus onde eles estão sendo treinados.

Além de se extrair relações linguísticas, muitos corpora também são montados para possibilitar o treinamento de modelos de aprendizado de máquinas em tarefas que queremos automatizar. Algumas tarefas estão relacionadas a anotações automáticas das palavras de um texto. Pode-se treinar um modelo para identificar as classes gramaticais das palavras e as relações de dependências sintáticas nas sentenças, assim como identificar entidades relevantes, como nomes de pessoas, empresas ou locais por exemplo. Essas marcações automáticas podem ser muito úteis para desambiguar um texto ou gerar metadados de forma automática. Marcações automáticas, em geral, são etapas preliminares do treinamento de tarefas mais complexas como classificação automática de textos, análise de sentimento ou até tarefas de perguntas e respostas.

A criação de corpora é uma atividade bem antiga e tem sido a base de muitos estudos linguísticos. Nas últimas décadas, com o crescimento dos recursos computacionais, cresceu muito a disponibilidade e o uso de corpora. Na tabela abaixo é listado alguns dos corpora mais utilizados.

|  |  |  |
| --- | --- | --- |
| Nome | Descrição | Link |
| Reuters News Dataset | Notícias da agência Reuters do ano de 1987. | https://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection |
| IMDB Reviews | Avaliações de 2.500 filmes para análise de sentimento binária | http://ai.stanford.edu/~amaas/data/sentiment/ |
| The WikiQA Corpus | Corpus com pares de perguntas e respostas em domínio geral. | https://www.microsoft.com/en-us/download/details.aspx?id=52419&from=http%3A%2F%2Fresearch.microsoft.com%2Fapps%2Fmobile%2Fdownload.aspx%3Fp%3D4495da01-db8c-4041-a7f6-7984a4f6a905 |
| Sentiment140 | Base com 160,000 *tweets* formatados com 6 atributos: polarização, ID, data do *tweet*, consulta, usuário e texto. Emoticons foram removidos. | http://help.sentiment140.com/for-students/ |
| Twitter US Airline Sentiment | *Tweets* sobre companhias aéreas americanas, recuperados em fevereiro de 2015, classificados como positivo, negativo ou neutro. | https://www.kaggle.com/crowdflower/twitter-airline-sentiment |
| Spoken Wikipédia Corpora | Centenas de horas de áudio contendo artigos da Wikipédia lidos em inglês, alemão e holandês. | https://nats.gitlab.io/swc/ |
| Wikipédia Links Data | 13 milhões de documentos composto por páginas web contendo pelo menos um hiperlink apontando para a Wikipédia em inglês. Cada página da Wikipédia é considerada uma entidade e os links representam uma menção as entidades. | https://code.google.com/archive/p/wiki-links/downloads |
| Gutenberg eBooks List | Lista anotada de ebooks do Projeto Gutenberg contendo as informações básicas dos livros e organizado por ano. | http://www.gutenberg.org/wiki/Gutenberg:Offline\_Catalogs |
| Jeopardy | O arquivo contém mais de 200.000 perguntas e respostas do programa Jeopardy. Cada ponto de dado possui outras informações complementares como categoria da pergunta, número do programa e data. | https://www.reddit.com/r/datasets/comments/1uyd0t/200000\_jeopardy\_questions\_in\_a\_json\_file/ |
| European Parliament Proceedings Parallel Corpus | Pares de sentenças de proveniente de processos do Parlamento Europeu em 21 idiomas europeus. | http://statmt.org/europarl/ |

Fonte: ALI, 2019

Em português, a disponibilidade de copora para treinamento de modelos para processamento de linguagem natural é muito menor do que em inglês ou outras línguas. Segue abaixo alguns exemplos dos corpora mais usados em português.

|  |  |  |
| --- | --- | --- |
| Nome | Descrição | Link |
| O Corpus do Português | Corpus contendo um bilhão de palavras em português. | https://www.corpusdoportugues.org/ |
| NILC / San Carlos Corpora | Vários corpora com texto em português contemporâneo, todos com POS-Tagging. | https://www.linguateca.pt/acesso/corpus.php?corpus=SAOCARLOS |
| *Tweets* em portugês para análise de sentimento | 800 mil *tweets* em português, divididos em positivo, negativo e neutro. | https://www.kaggle.com/augustop/portuguese-tweets-for-sentiment-analysis |
| BlogSet-BR | Um extensivo corpus com 2.1 bilhões de palavras provenientes de 808 blogs e 7,4 milhões de posts escritos em português brasileiro. | http://www.inf.pucrs.br/linatural/wordpress/recursos-e-ferramentas/blogset-br/ |

Fonte: NGUYEN, 2019

Finalmente, para muitas aplicações é mais interessante trabalhar com corpora de domínio específicos do que de domínio geral. É possível extrair termos técnicos e jargões de documentos de uma área específica que dificilmente apareceriam em textos de amplo domínio. Da mesma maneira, algumas palavras possuem significados ou são usadas diferentemente em áreas do conhecimento distintas. Dois exemplos de corpora da área da biologia são o *BioCreative*[[2]](#footnote-2) e o GENIA[[3]](#footnote-3).

Neste trabalho, o objetivo foi criar um corpus monolíngue especializado no domínio de óleo e gás. Uma rica fonte de textos técnicos, de diversas áreas do conhecimento e públicos são os repositórios e bibliotecas de instituições de pesquisas e universidades. Por esse motivo, teses e dissertações foram os documentos que serviram de fonte de textos para o corpus criado.

# Teses e Dissertações em Óleo e Gás

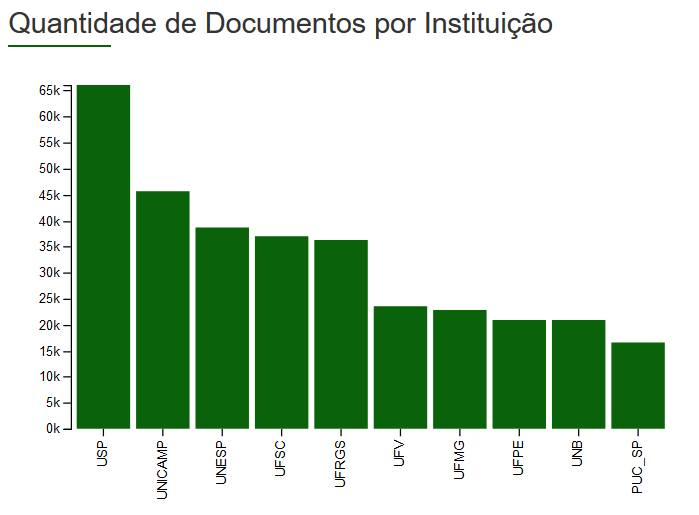
Para preencher a lacuna que existe na disponibilidade de um corpus público em português no domínio da indústria do petróleo, este trabalho focou em criar um corpus composto por teses e dissertações acadêmicas. Para esse fim, utilizamos como fonte a Base Digital de Teses e Dissertações (BDTD)[[4]](#footnote-4) do Instituto Brasileiro de Informação de Ciência e Tecnologia (IBICT). Nessa base estão disponíveis quase 600.000 documentos das principais instituições de pesquisa do Brasil.

A escolha das teses e dissertações para serem os componentes básicos desse corpus se deu pelos seguintes motivos. Primeiramente, teses e dissertações são resultados de pesquisas de mestrado e doutorados e são públicas. Essa característica facilita o compartilhamento desse corpus entre diversos grupos de pesquisa e torná-lo uma referência entre os corpora em português. Por serem documentos técnicos, esse tipo de documento possui uma variedade de termos e jargões específicos das áreas de conhecimento a que pertencem. Uma outra vantagem é o fato de serem documentos relativamente bem padronizados, as diagramações das páginas são relativamente simples o que facilita a extração dos textos dos documentos. Por fim, teses e dissertações são documentos longos quando comparados à documentos presentes em outros corpora, como por exemplo *tweets*, verbetes da Wikipédia e artigos jornalísticos.

A principal desvantagem da utilização das teses e dissertações está no fato desses documentos estarem no formato PDF (*Portable Document Format*). Buscar e extrair informações rápida e eficientemente de documentos PDF ainda é um problema a ser resolvido, esta é uma das principais áreas de pesquisa do ramo conhecido como Extração da Informação (EI) (Yeung, 2005). Outra desvantagem de um corpus formado por documentos acadêmicos está no fato dele conter jargões e estilos de escrita particulares da academia. Pode haver alguma incompatibilidade entre a linguagem utilizada nas teses e dissertações quando comparado aos documentos utilizados no dia a dia da indústria.

A Base Digital de Teses e Dissertações (BDTD) é referência no acesso livre ao conhecimento e reconhecida por ser uma das maiores bibliotecas do mundo em número de teses e dissertações de um só país. A origens do Instituto Brasileiro de Informação de Ciência Tecnologia remonta à década de 50, quando a UNESCO incentivou a criação no Brasil de um centro nacional de bibliografia. Em 2002, é criada a BDTD e hoje integra o sistema de informação de teses e dissertações das instituições de ensino e pesquisa brasileiras. (IBICT, 2018)

Em 2019, a BDTD contava com 586.225 documentos cadastrados em sua base, provenientes de 116 instituições diferentes. Desses documentos, 431.225 eram dissertações de mestrado e 155.002 eram teses de doutorado. Anualmente, são depositados cerca de 40 mil novos documentos, em sua grande maioria em português. As universidades públicas, tanto federais quanto estaduais, são as grandes depositárias da BDTD. (IBICT, 2019)

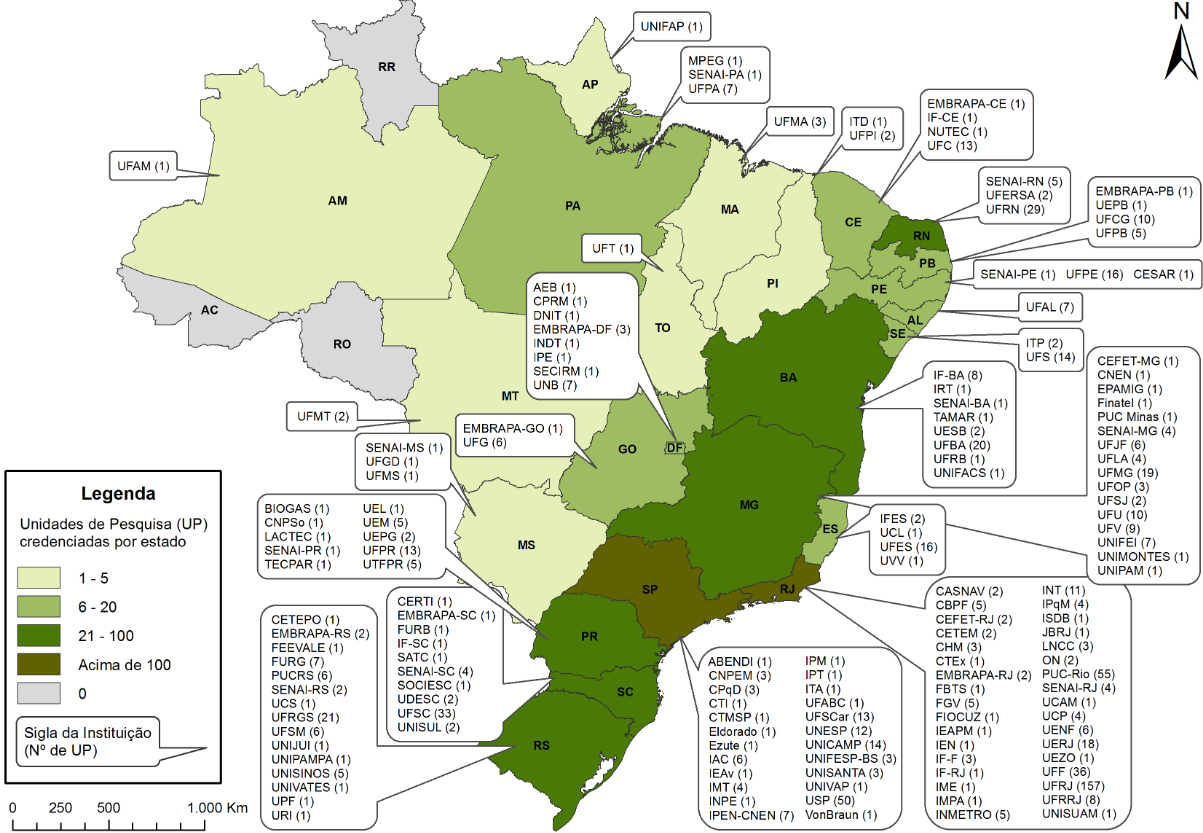


Quantidade de documento por instituição (IBICT, 2019)

O corpus de óleo em gás, portanto, foi composto por documentos cadastrados na BDTD. Primeiramente, foi necessário identificar, dentre todas as teses e dissertações existentes, quais eram as mais relevantes para montar um corpus em português de óleo e gás. No entanto, a BDTD só possui os resumos das teses e dissertações, os documentos completos estão nos repositórios de cada uma das instituições. Portanto, seria muito difícil buscar de forma automática em todas as 116 instituições presentes da BDTD.

A estratégia utilizada foi escolher as instituições que teriam a maior chance de ter produzido teses e dissertações na área de conhecimento alvo. O primeiro critério utilizado foi escolher as 9 instituições com mais documentos depositados na BDTD. Os destaques, segundo esse critério, foram USP e UNICAMP com mais de 45 mil documentos cada uma, e UNESP, UFSC e UFRGS com quase 40 mil cada. (IBICT, 2019)

O segundo critério utilizado foi escolher instituições com Unidades de Pesquisas na área de óleo e gás. Para tal, foi utilizado o credenciamento feito pela Agência Nacional de Petróleo, Gás Natural e Biocombustível (ANP) às instituições de pesquisa e desenvolvimento. Esse credenciamento é o reconhecimento formal da agência reguladora de que a unidade de pesquisa possui condições técnicas e infraestrutura necessária para a execução de projetos com recursos da clausula de PD&I[[5]](#footnote-5). Até setembro de 2019, foram credenciadas 150 instituições que, em conjunto, possuíam 898 unidades de pesquisas nas áreas de interesse da ANP. Escolhemos as instituições que possuíam mais de 10 unidades de pesquisas credenciadas à ANP. (ANP, 2019)

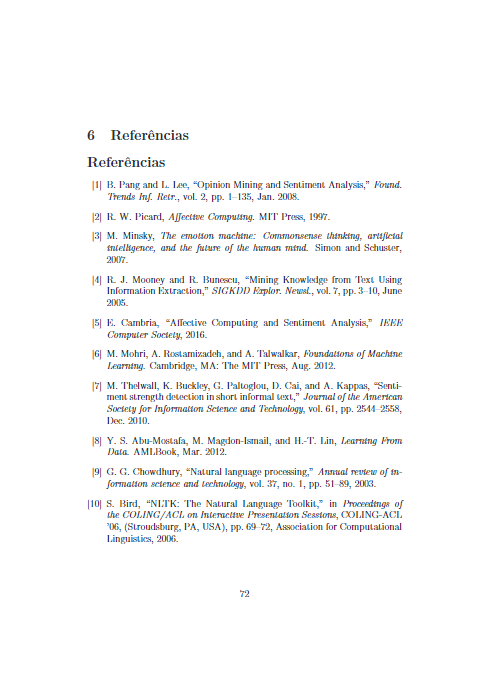
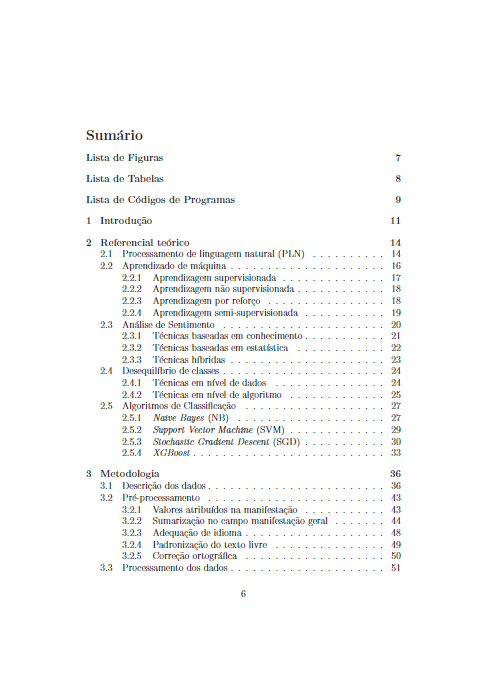
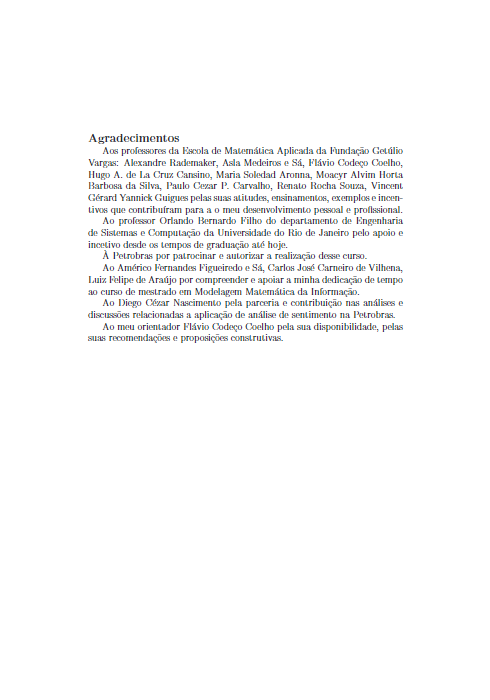
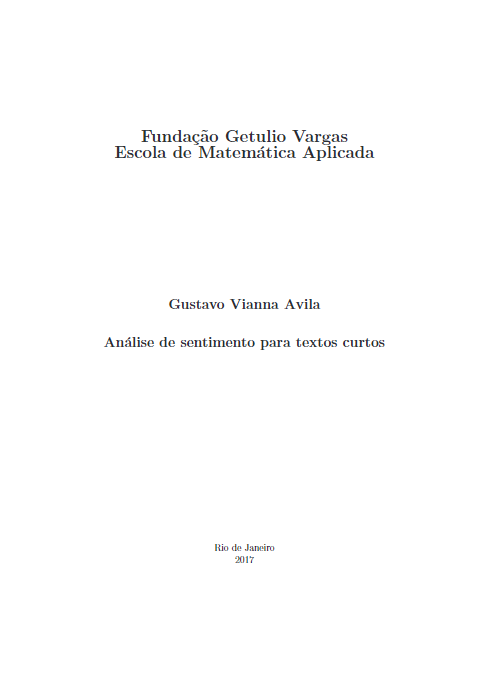


Instituições Credenciadas pela ANP (até 30 de setembro 2019) (ANP, 2019)

Por fim, algumas instituições que se enquadravam nos critérios levantados não puderam ser incluídas devido a dificuldades técnicas em recuperar os documentos dos seus repositórios institucionais. Dois casos notáveis foram a UFRJ e PUC-Rio. Essas duas instituições, por exemplo, possuíam repositórios em plataformas que dificultavam a recuperação automática ou exigiam *login* e senha para acessar. Em suma, as instituições consideradas foram:

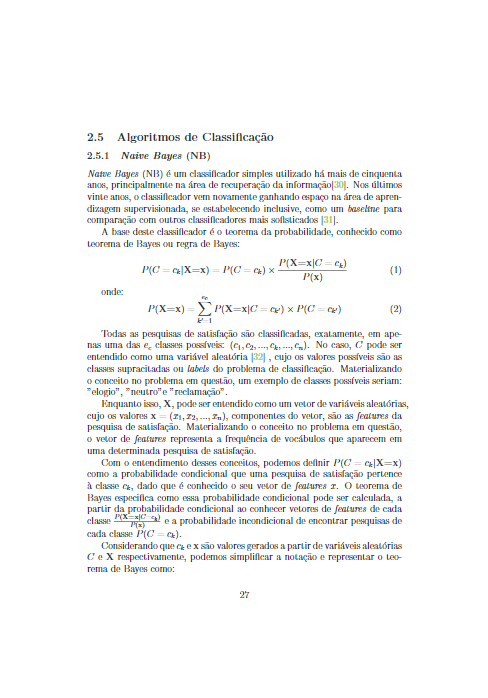
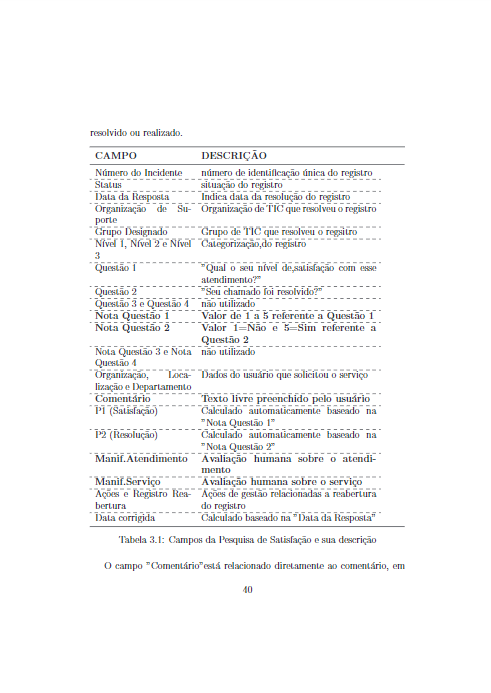
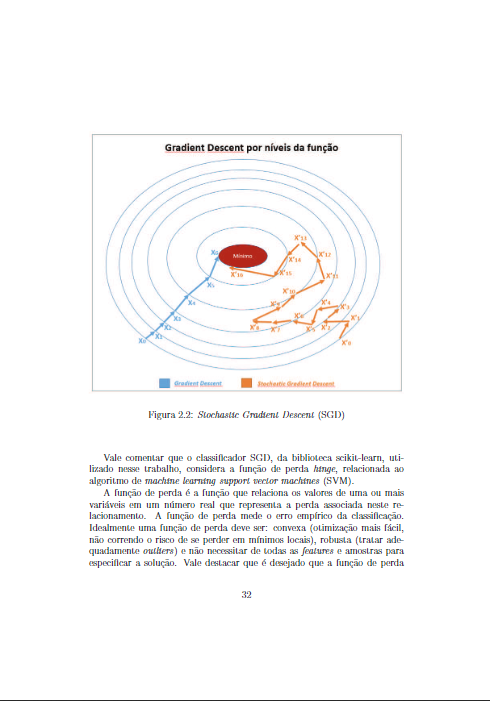
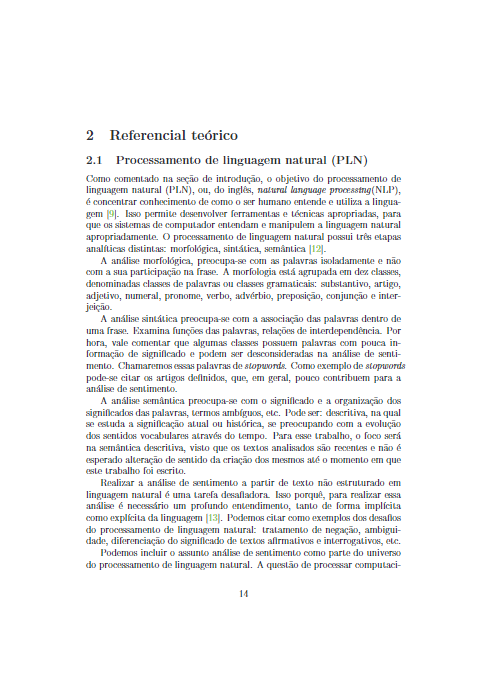
* UFBA – Universidade Federal da Bahia
* UFES – Universidade Federal do Espirito Santo
* UFF – Universidade Federal Fluminense
* UFMG – Universidade Federal de Minas Gerais
* UFPE – Universidade Federal de Pernambuco
* UFRGS – Universidade Federal do Rio Grande do Sul
* UFRN – Universidade Federal do Rio Grande do Norte
* UFS – Universidade Federal de Sergipe
* UFSC – Universidade Federal de Santa Catarina
* UFSCar – Universidade Federal de São Carlos
* UFV – Universidade Federal de Viçosa
* UnB – Universidade de Brasília
* UNESP – Universidade Estadual Paulista
* UNICAMP – Universidade Estadual de Campinas
* USP – Universidade de São Paulo

O objetivo final do corpus é a utilização dos textos das teses e dissertações para tarefas de processamento de linguagem natural. No entanto esses documentos possuem outras informações além de textos puros. Em geral, uma tese ou dissertação é composta por algumas páginas iniciais contendo capa, agradecimentos e sumários, da mesma forma as últimas páginas são compostas por referências bibliográficas e anexos. Apesar de conter texto, o conteúdo dessas páginas costuma ser ruim para a montagem do corpus por possuírem pouco texto corrido ou textos com pouca relevância.



Páginas iniciais e finais: Capa, agradecimentos, sumários e bibliografia (Avila, 2017)

Os capítulos dos documentos são o alvo da montagem do corpus. Neles são encontrados os principais termos técnicos e a maior quantidade de texto. Entretanto, os capítulos não são formados exclusivamente por textos, neles também são encontradas muitas figuras, tabelas e fórmulas. Apesar das figuras, tabelas e fórmulas serem importante para a leitura, para a montagem de um corpus esses elementos podem atrapalhar diversos algoritmos de aprendizado de máquina. A extração das tabelas dos documentos, por exemplo, pode resultar em um texto ruidoso, onde o texto do capítulo acaba misturado com as linhas das tabelas. Neste caso, é mais interessante eliminar a tabelas e utilizar somente o texto para o treinamento. O mesmo pode ocorrer com as fórmulas e figuras.



Conteúdo dos capítulos: Texto, figuras, tabelas e fórmulas (Avila, 2017)

Por fim, além do conteúdo do documento propriamente dito, também faz parte do corpus informações referentes aos documentos, ou seja, os metadados. Para as teses e dissertações extraídas da BDTD, é possível recuperar os seguintes metadados:

* Título
* Nível de acesso, em geral aberto
* Data de defesa
* Autor/a
* Orientador/a
* Coorientador/a
* Tipo de documento, tese ou dissertação
* Idioma, maioria em português
* Instituição de defesa
* Programa
* Assuntos em português
* Áreas de conhecimento
* Link para download do texto completo
* Resumo em português:
* Resumo em inglês
* Banca

A escolha da BDTD proporcionou uma farta fonte de documentos, no entanto, também acrescentou o desafio de escolher aqueles mais relevantes. Por esse motivo, a classificação das teses e dissertações foi uma das etapas mais importantes da criação de um corpus de óleo e gás.

# Classificação de textos

A Base Digital de Teses e Dissertações (BDTD) do IBICT é um repositório que contém documentos de todas as áreas do conhecimento. Portanto, foi necessário selecionar quais documentos eram relevantes para entrar no corpus de óleo e gás e quais deveriam ser deixados de fora. A maneira mais direta de realizar essa tarefa seria solicitar que especialistas lessem os resumos dos documentos e decidissem se eles eram relevantes ou não para a indústria do petróleo. No entanto essa tarefa seria extremamente exaustiva e demorada. A abordagem utilizada, portanto, foi a utilização de algoritmos de classificação de textos para realizar essa tarefa.

A classificação de textos, também conhecida por categorização ou identificação de tópicos, é a tarefa de classificar um texto aleatório no seu respectivo domínio. Essa classificação pode ser binária, ou seja, identificar se um dado texto pertence ou não a determinado domínio, ou multi classe, quando há diversas classes possíveis. Os problemas de classificação também podem ser divididos em *single* ou *multi-label*, quando são atribuídas, respectivamente, um ou várias classes para cada texto. (Obaidullah, 2019)

Para a realização das tarefas de *text minning* e de processamento de linguagem natural, primeiramente, é necessário extrair os atributos que serão utilizados pelos algoritmos de aprendizado de máquina. Em geral, as palavras são utilizadas como unidades básicas a serem tratadas como atributos dos documentos. As palavras podem ser tratadas isoladamente, uni-gramas, ou em conjunto, bi-gramas, tri-gramas ou n-gramas. Outros elementos do texto também poderiam ser tratados considerados atributos, como letras, números, caracteres especiais, sentenças e parágrafos.

Uma forma clássica de representar um texto é conhecida por *bag-of-words*. Neste tipo de representação, cada documento é representado como um vetor. O tamanho desse vetor é igual ao tamanho do vocabulário e os valores representam a frequência que cada palavra aparece no documento (Brownlee, 2017). Outra representação bastante utilizada é a TF-IDF (*Term Frequency – Inverse Document Frequency*). Nele o valor de cada palavra, ou n-gram, é dado pela sua exaustividade, o quão frequente um termo se apresenta em um documento (TF), e especificidade, o quão raro esse termo aparece na coleção de documentos (IDF) (Baeza-Yates e Ribeiro-Neto, 2013). Assim como na representação *bag-of-words,* cada documento pode ser representado por um vetor do tamanho do vocabulário, cujos valores são os índices TF-IDF de cada termo para cada documento.

Uma das dificuldades das representações *bag-of-words* e TF-IDF é a necessidade de se trabalhar com longos vetores esparsos, ou seja, os vetores podem ter dezenas de milhares de valores representados, mas quase todos são zeros. Outra característica é a ausência de relação entre os termos, nem a ordem em que as palavras aparecem no texto nem a similaridades entre as palavras são capturadas por essas representações. Esses são justamente os problemas atacados pela representação de palavras por vetores obtidas a partir de treinamento de redes neurais, também conhecida por Word2Vec (Mikolov et. Al, 2013). No Word2Vec, bem como em outras formas de *word embeddings*, cada palavra é representada por um vetor de tamanho predefinido e treinados previamente.

Para a classificação dos textos foram testados diversos algoritmos de aprendizado de máquina. Primeiramente, foram utilizados algoritmos clássicos como *Naive Bayes,* Regressão Logística, *Support Vector Machine, Bagging* e *Boosting Models*. Em seguida, também foram testados os algoritmos baseados em redes neurais, tanto redes rasas quando redes profundas, como as redes convolucionais, *Long-Short Term Model* (LSTM), *Gated Rated Unit* (GRU), bidirecionais e redes convolucionais recorrente.

Dentre os algoritmos clássicos, o primeiro algoritmo testado foi o *Naive Bayes*. É o algoritmo baseado no teorema de Bayes e é particularmente recomendado para alta dimensionalidade. Em seguida, foi a vez da Regressão Logística, método onde as probabilidades dos resultados são descritas como uma função das variáveis observadas. Já o *Support Vector Machine* é um algoritmo que encontra um hiperplano que separa linearmente os dados de treinamento de acordo com as classes apresentadas. Finalmente, os modelos de *Bagging* e *Boosting* agrupam diversos classificadores mais simples para gerar um classificador mais preciso. (Aggarwal, 2014)

Já para os classificadores baseados em redes neurais, foram utilizadas tanto as redes com poucas camadas, as chamadas *multi layer perceptron,* como os algoritmos de *deep learning*. Dos modelos de *deep learning* o primeiro a ser testado foi a rede convolucional, que faz uso de camadas convolucionais para extrair diferentes atributos dos dados. Em seguida, foram testados alguns modelos de redes neurais recorrentes (RNN), ou seja, modelos que levam em consideração a sequência de entrada dos dados, algo muito importante em textos. Foram testadas duas arquiteturas de redes recorrentes a *Long-Short Term Model* (LSTM) e a *Gated Rated Unit* (GRU). Também foi testada uma rede bidirecional, arquitetura de rede recorrente que usa os dois sentidos possíveis de entrada do texto. Por fim, também foi testado uma arquitetura tanto com camadas convolucionais quanto com camadas recorrentes (RCNN). (Ravichandiran, 2019)

Uma vez que a fonte de documentos e os algoritmos de classificação estavam disponíveis, foi possível realizar os experimentos para treinar o melhor classificador. Por sua vez, o classificador foi o bloco fundamental para a criação do corpus de óleo e gás.

# Metodologia

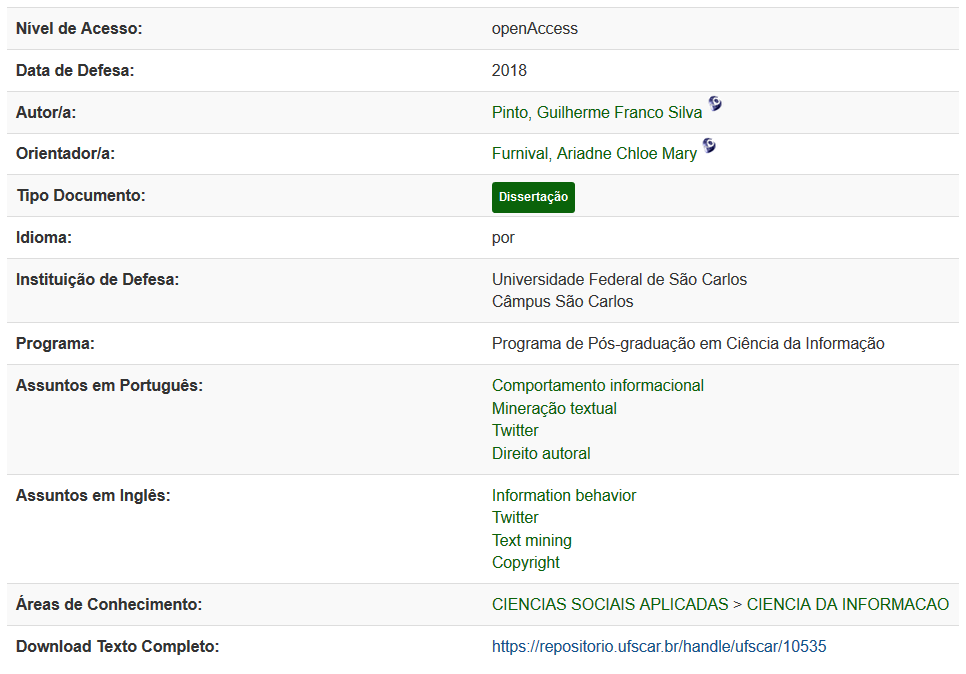
Para a criação de um corpus de textos em português do domínio específico de óleo e gás foram necessárias diversas etapas. Inicialmente, foi preciso ler os resumos das teses e dissertações presentes na BDTD. Portanto, foi necessário criar um *crawler[[6]](#footnote-6)* capaz de acessar automaticamente esses documentos. Em seguida, um algoritmo classificador de texto foi treinado para reconhecer se um determinado resumo pertencia a um documento relevante ao setor de petróleo. Com esse classificador em mãos, foi possível utilizar o *crawler* para ler todos os documentos das instituições selecionadas e identificar se eles eram relevantes ou não para a montagem do corpus. Uma vez identificados os documentos relevantes, foi criado um outro conjunto de *crawlers*, um para cada instituição de pesquisa, com o objetivo de coletar os arquivos originais das teses e dissertações nos seus respectivos repositórios institucionais. Finalmente, após todos os documentos coletados, foi possível extrair apenas os textos dos documentos para se criar um corpus do domínio de óleo e gás em português.

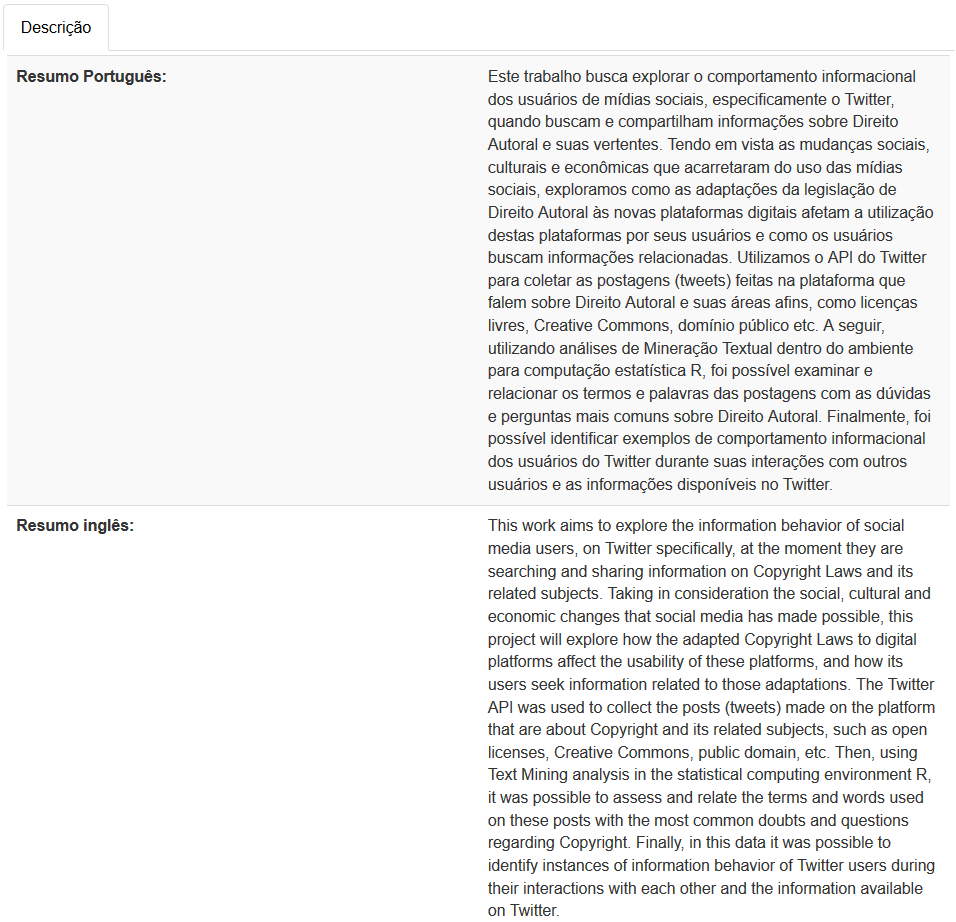
Etapas para a criação de um corpus de textos no domínio de óleo e gás em português.

## Crawler da BDTD

A primeira tarefa realizada foi a criação de um *script* que possibilitasse a coleta automática dos metadados de um documento presente na BDTD. Cada documento é apresentado em uma página com uma URL[[7]](#footnote-7) própria. A primeira parte da URL sempre começa com <http://bdtd.ibict.br/vufind/Record>, já a última parte é composta por um identificador único do documento na BDTD.

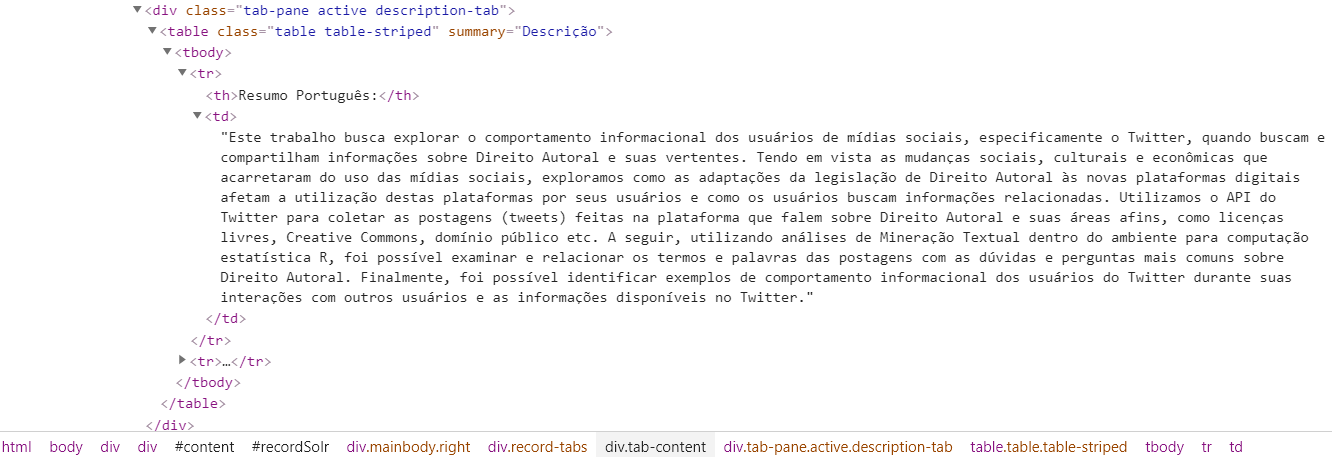
Cada página de uma tese ou dissertação possui uma tabela com metadados como a do exemplo abaixo:





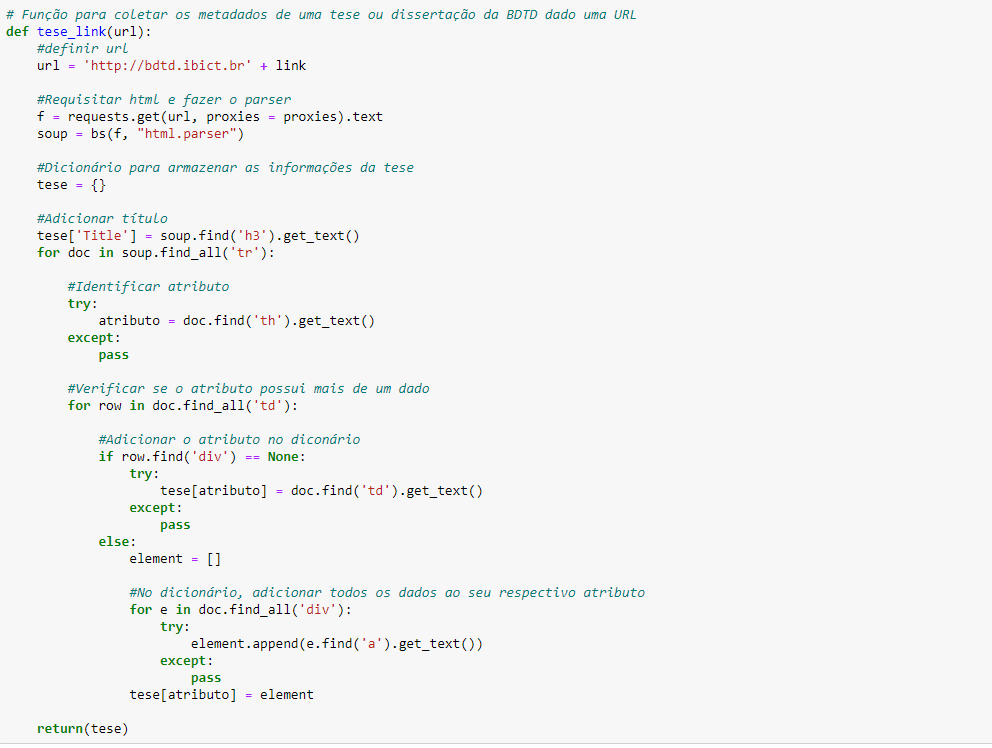
Exemplo de metadados no website da BDTD (Pinto, 2018)

É possível recuperar os metadados das tabelas com o auxílio das suas respectivas marcações HTML[[8]](#footnote-8).



Exemplo do código HTML da dissertação de Pinto (2018) na BDTD

Para a criação do *script* de recuperação dos metadados da BDTD foram usadas, principalmente, duas bibliotecas em Python, *Request* e *BeautifulSoap*. Essas bibliotecas são respectivamente para acessar páginas de internet e analisar documentos em formatos HTML, XML[[9]](#footnote-9) e JSON[[10]](#footnote-10), dentre outros. Por fim, para recuperar todos os metadados de um documento, dada uma URL específica, foi criado a seguinte função:



Função para coletar metadados de um documento na BDTD

Finalmente, esses metadados foram armazenados em um dicionário para posterior exportação em formato JSON.

## Classificador de textos

Para criar um algoritmo capaz de receber um texto aleatório e ser capaz de identificar se ele é relevante para o domínio de petróleo, primeiramente, foi preciso ter uma amostra relevante de exemplos para treinamento. Foi necessário, portanto, criar uma base de treinamento com textos classificados como relevantes e não relevantes.

Para a criação dessa base de treinamento utilizamos uma estratégia para construir o conjunto de documentos relevantes e outra estratégia para os documentos não relevantes ao setor de petróleo. Para montar a base de documento relevantes foi utilizada uma abordagem bem direta, buscar na BDTD os documentos cujo assunto continha os termos “petroleo” e “gas natural”. Através dessa busca foi possível recuperar 1.986 documentos. Uma abordagem complementar foi a identificação de funcionários da Petrobras[[11]](#footnote-11) que haviam depositado alguma tese ou dissertação no seu repositório institucional. Através dessa lista, buscou-se na BDTD os documentos desses autores. Com isso foi possível acrescentar mais 504 documentos ao conjunto de teses e dissertações do domínio de óleo e gás.

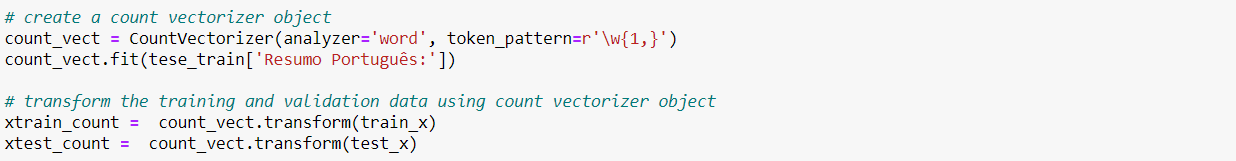
Para montar a base de teses e dissertações que não fazem parte do domínio de óleo e gás foi utilizada as áreas do conhecimento do CNPQ[[12]](#footnote-12), um dos metadados presentes na BDTD. Para manter a abrangência em diversos tipos de textos, foi escolhido 200 teses de cada área do conhecimento. Desta forma, foi montada uma base de treinamento com 2.490 documentos dentro do domínio de óleo e gás e 11.532 documentos de outras áreas.

O pré-processamento e a classificação foram baseados no trabalho de Bansal (2018). Primeiramente, todos os textos foram limpos e pré-processados. Em seguida, foram gerados os atributos de entradas necessários para os algoritmos de classificação. Com os atributos disponíveis, diversos classificadores foram treinados e avaliados.

O pré-processamento e a engenharia de atributos são as etapas na qual o texto é preparado para ser usado pelos algoritmos de classificação. Inicialmente, foi verificado se os documentos extraídos, usando as diversas estratégias de coleta na BDTD, não se repetiam. A etapa seguinte foi a separação entre os textos em português e em outro idioma, em geral inglês. Apesar da grande maioria dos documentos da BDTD estarem em português, há uma pequena parcela em inglês. Um fato bastante comum encontrado nos documentos foi a concatenação dos textos dos campos “Resumo Português” e “Resumo Inglês”. A mistura dos dois idiomas “sujava” os textos, piorando a acurácia do classificador. Foi necessário a utilização da biblioteca *Langdetect[[13]](#footnote-13)* para separa os resumos. Os textos sem resumos em português foram excluídos.

Na etapa seguinte, foram feitas as alterações no texto propriamente dito. Todas as letras foram deixadas em minúscula, as acentuações e as *stopwords* foram extraídas. A base de treinamento ficou com um grande desbalanceamento entre as duas classes. Portanto, foram sorteados para treinamento uma quantidade de documentos que deixasse o conjunto de treinamento balanceado. Desta forma, foram usados 4.242 resumos, 2.121 textos de cada classe. Destes, 80% foram usados para o treinamento dos modelos de classificação e os 20% restantes serviram para a avaliação.

Após os textos serem limpos e tratados, foram utilizadas algumas técnicas de extração de atributos dos textos. A primeira, e de certa forma a mais simples, foi transformar os textos em um vetor de frequência*.* Também foi utilizada a transformação do texto para vetores TF-IDF. Foram gerados três tipos de vetores TF-IDF, um com a representação de cada palavra, outro com a representação dos bigramas e trigramas, e finalmente, uma última representação no nível dos caracteres. Todas as transformações foram feitas utilizando a biblioteca *Sklearn*[[14]](#footnote-14).

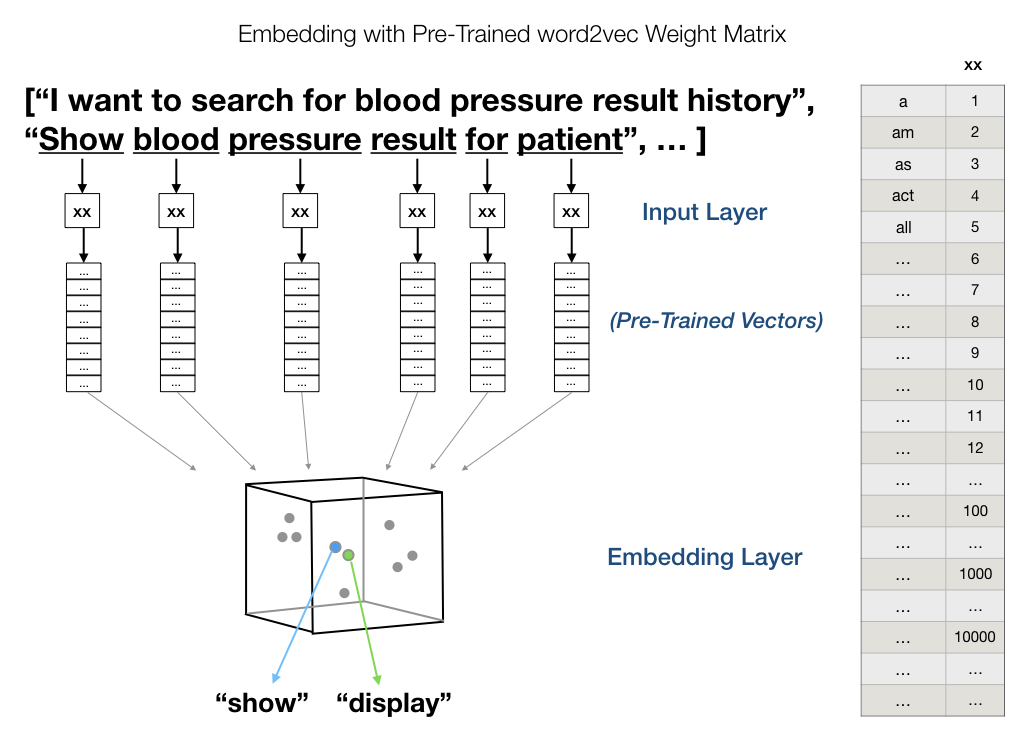


Criando os vetores de frequência de palavras

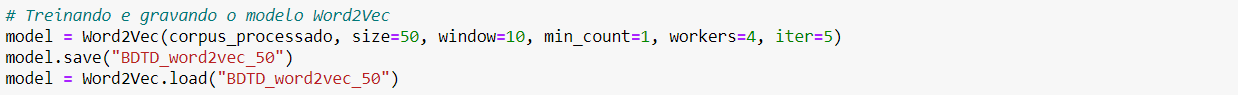


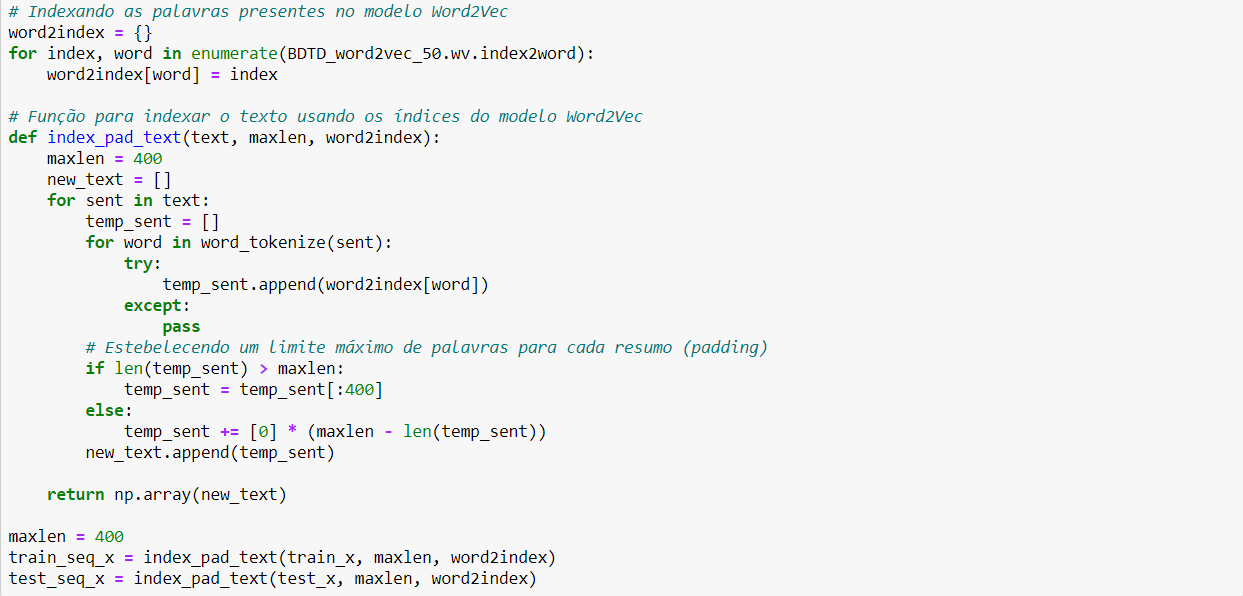
Criando os vetores TF-IDF

A última transformação de texto para vetores foi feita utilizando o algoritmo *Word2Vec*. Primeiramente, foi treinado um modelo vetorial de palavras usando todos os textos disponíveis, ou seja, não foi necessário ter uma base balanceada. Com isso obtivemos um modelo onde era possível mapear cada palavra do dicionário para um vetor correspondente. Esses vetores foram utilizados para representar os documentos.



Representação de textos usando vetorização de palavras. (Mangiavacchi, 2018)





Treinando e indexando os textos de acordo com o modelo Word2Vec

Com os atributos criados, era possível treinar os modelos de classificação usando diversas combinações de algoritmos e atributos. Para tal, foi criada uma função que permitisse entrar com os atributos e os tipos de classificadores. A função disparava o treinamento e calculava a acurácia usando a base de teste.



Função para treinar modelo de classificação de textos

Foram treinados 29 classificadores diferentes usando os cinco possíveis atributos. Para os algoritmos clássicos de aprendizado de máquina, foram usados como entrada os vetores *CountVector* e os três TF-IDF. Já para os algoritmos de *deep learning*, foi usado o Word2Vec. O melhor classificador encontrado foi o *Suport Vector Machine* com acurácia de 92,80%.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Classificador** | **CountVector** | **TFIDF Palavra** | **TFIDF Ngram** | **TFIDF Caractere** | **Word2Vec** |
| Naive Bayes | 0,7405 | 0,7275 | 0,8926 | 0,8360 |  |
| Logistic Regression | 0,7428 | 0,7417 | 0,9198 | 0,7971 |  |
| Suport Vector Machine | 0,7535 | 0,7452 | 0,9280 | 0,8478 |  |
| Random Forest | 0,6898 | 0,7004 | 0,8561 | 0,8042 |  |
| Xtereme Gradient Boosting | 0,7358 | 0,7476 | 0,9221 | 0,8867 |  |
| Multi Layer Perceptron | 0,7665 | 0,7346 | 0,9186 | 0,8832 |  |
| Convolutional NN |  |  |  |  | 0,8620 |
| Recurrent NN - LSTM |  |  |  |  | 0,8820 |
| Recurrent NN - GRU |  |  |  |  | 0,8726 |
| Bidirectional Recurrent NN |  |  |  |  | 0,8620 |
| Recurrent Convolutional NN |  |  |  |  | 0,8679 |

Acurácia dos modelos de classificação treinados



Treinamento dos algoritmos Suport Vector Machine, sua acurácia e matriz de confusão

## Teses relevantes para o domínio óleo e gás

Na etapa anterior, foi criado um conjunto de documentos para o treinamento de modelos que fossem capazes de classificar se um texto é relevante ou não para o domínio de óleo e gás. Com esses modelos classificadores em mãos, foi possível ler centenas de milhares de documentos de diversas instituições e decidir quais deveriam compor o corpus de óleo e gás. Primeiramente, foi utilizado o mesmo *crawler* da BDTD para ler todos os resumos de teses e dissertações. Para aqueles documentos que se mostrassem relevantes, foram construídos novos *crawlers* para osrepositórios das instituições de pesquisa para fazer o *download* dos arquivos.

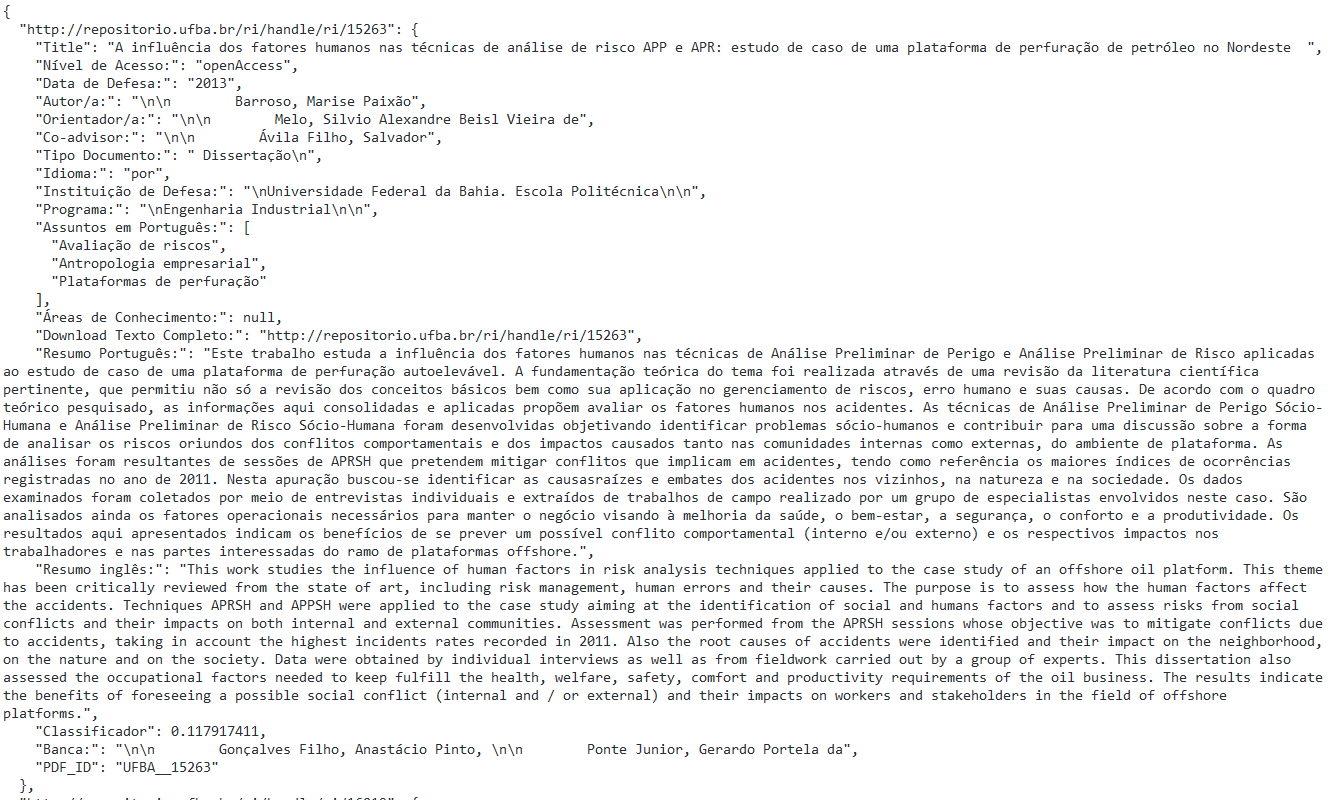
A busca pelas teses relevantes foi feita uma instituição por vez. Por exemplo, inicialmente buscou-se na BDTD todas os documentos da UFBA[[15]](#footnote-15). Foram recuperados quase 10.000 *links* com resumos de documentos dessa instituição. Esses resumos tiveram que ser pré-processados da mesma maneira que foram processados os textos usados no treinamento. Cada um desses resumos foi passado pelo classificador que fez uma predição quanto a relevância ao domínio de petróleo. O modelo classificou cada texto com um valor entre 0 a 1, onde 0 eram os mais relevantes para o domínio do petróleo. Teoricamente, poderiam ser recuperados todos os documentos com predição abaixo de 0,5. No entanto, buscando ser conservador e ter certeza que os documentos recuperados pertenciam ao domínio de interesse, foram recuperados apenas aqueles com predição abaixo de 0,2. Usando esse procedimento para todas as 15 instituições foram recuperados 4.467 teses e dissertações. Ou seja, de todos os documentos disponíveis na BDTD, apenas uma pequena fração era relevante para fazer parte do corpus de óleo e gás.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Total de Teses** | **Teses recuperadas** | Percentual de teses recuperadas sobre o total |
| **Total** | **358,942** | **4,467** | **1.24%** |
| USP | 66,077 | 685 | 1.04% |
| UNICAMP | 45,644 | 935 | 2.05% |
| UNESP | 38,008 | 291 | 0.77% |
| UFSC | 36,971 | 308 | 0.83% |
| UFRGS | 35,665 | 471 | 1.32% |
| UFV | 23,446 | 44 | 0.19% |
| UFMG | 22,060 | 146 | 0.66% |
| UNB | 20,780 | 29 | 0.14% |
| UFPE | 20,551 | 386 | 1.88% |
| UFRN | 12,625 | 587 | 4.65% |
| UFSCAR | 11,076 | 97 | 0.88% |
| UFBA | 9,916 | 186 | 1.88% |
| UFES | 9,720 | 204 | 2.10% |
| UFS | 4,806 | 71 | 1.48% |
| UFF | 1,597 | 27 | 1.69% |

Documentos recuperados por instituição

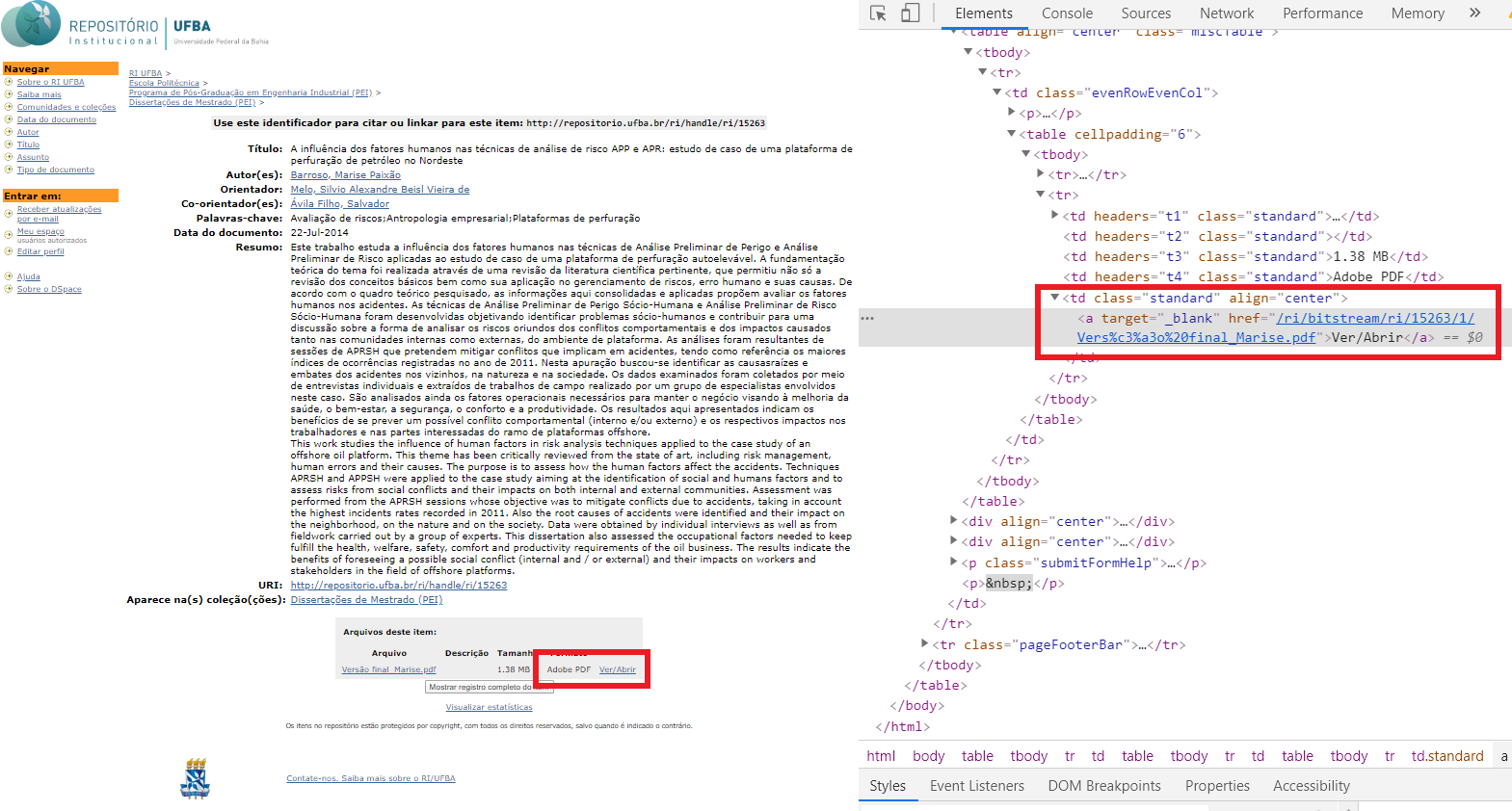
## *Crawler* para os repositórios institucionais

Uma vez identificados os documentos relevantes para a montagem do corpus, foi necessário recuperar os metadados presentes na BDTD e o arquivo com o texto integral depositado nos repositórios de cada instituição. Os metadados foram agrupados em um dicionário e gravados em um arquivo em formato JSON.

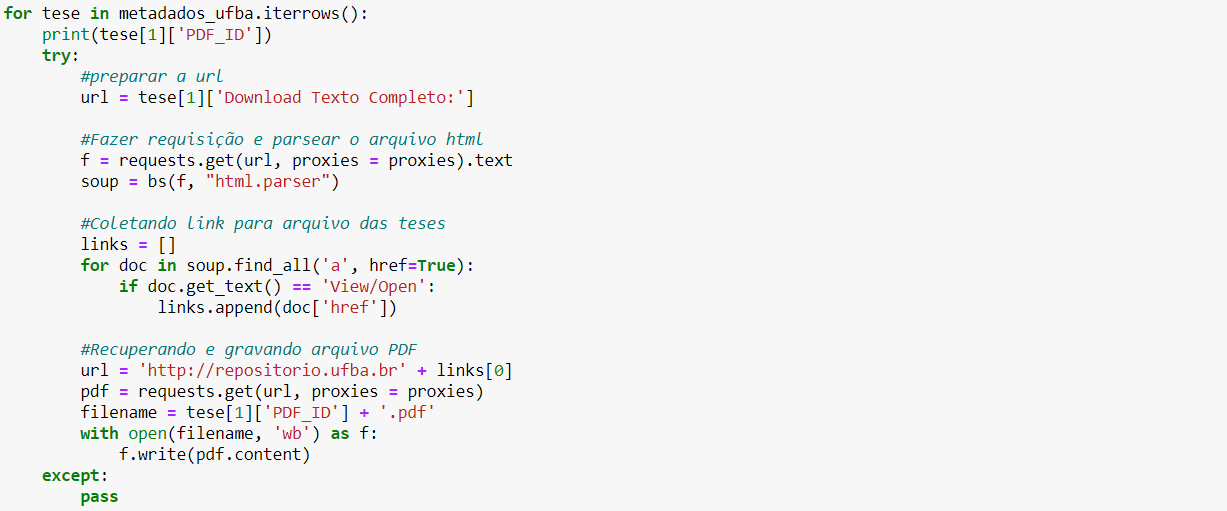


Exemplo de metadados de uma dissertação gravado em formato JSON

Já para a recuperação do arquivo completo foi criado um *crawler* para o repositório de cada uma das 15 instituições. O *link* para o documento completo é um dos metadados presentes na BDTD. No entanto, para cada repositório, foi necessário identificar a *tag html* específica que permitiria acessar automaticamente o documento completo. No caso da UFBA, por exemplo, o arquivo poderia ser encontrado procurando pelo texto “'View/Open” em todas as tags “a”. Desta forma, foi possível recuperar automaticamente todos os 186 documentos desta instituição.



Identificando *tag* para recuperar documento no repositório institucional da UFBA



*Crawler* para recuperar todas os documentos do repositório institucional da UFBA

## Extração de informações dos documentos

Uma vez que os documentos relevantes estavam disponíveis, a próxima etapa era efetivamente montar um corpus de óleo e gás. Para isso, foi necessário extrair os textos dos documentos em formato PDF. A solução utilizada para a extração dos textos dos arquivos foi a ferramenta Apache Tika[[16]](#footnote-16). Essa é uma das ferramentas mais populares para extração texto, pois ela suporta diversos formatos de arquivos, inclusive PDF. Muitas aplicações, como diversos motores de busca, utilizam o Apache Tika como solução de extração de texto.

No entanto, apesar de ser muito popular, o texto extraído utilizando o Apache Tika ainda apresenta algumas inconsistências. Por exemplo, textos oriundos de tabelas são convertidos como textos comuns e misturados com o restante do texto. O mesmo ocorre com legendas e textos presentes dentro das figuras. Desta forma, o texto final que irá compor o corpus de óleo e gás poderá conter algum ruído indesejado.

Para mitigar esses problemas, uma segunda etapa deste trabalho já está sendo planejada. Uma segunda extração do texto será realizada utilizando uma ferramenta de extração própria. Essa ferramenta usará técnicas de visão computacional, mais especificamente redes neurais convolucionais, para identificar as partes do documento onde há apenas texto. Com isso, será evitado a extração de elementos irrelevantes dos documentos, como sumários, referências bibliográficas, tabelas, figuras e fórmulas.

Os textos extraídos dessas 4.302 teses e dissertações compuseram o corpus final de óleo e gás. Esse corpus tem 74.445.798 palavras em 5.999.496 sentenças, sendo o maior corpus do domínio de óleo e gás na língua portuguesa.

# Conclusão

Neste trabalho foi criado um corpus de óleo e gás em português para ser utilizado nas inúmeras tarefas de processamento de linguagem natural. Esse corpus é o maior corpus desse domínio disponível em língua portuguesa. A vantagem de se ter um corpus em um domínio específico é a possibilidade de se capturar jargões técnicos e usos específicos de palavras que dificilmente seria encontrado em textos de domínio geral.

Para a montagem deste corpus, foi necessário criar dois tipos de robôs de extração de informações de páginas da internet, os chamados *crawlers*. O primeiro buscou informações no repositório da Base Digital de Teses e Dissertações do IBICT. Essa base possui informações sobre as principais instituições de pesquisas brasileiras. O segundo tipo de *crawler* recuperava os arquivos completos das teses e dissertações nos repositórios de cada instituição.

No entanto, de todas as teses e dissertações produzidas, apenas 1,24% são relevantes para a indústria de óleo e gás. Para diferenciar os documentos relevantes dos não relevantes, foi treinado um modelo baseado em aprendizado de máquina para fazer a classificação dos documentos.

Por fim, foi possível ler automaticamente o resumo de mais de 350 mil teses e dissertações e baixar apenas as 4.302 mais relevantes. Esses documentos serviram de base para um corpus com quase 75 milhões de palavras e que servirá de referência para os futuros trabalhos de processamento de linguagem natural na indústria do petróleo.

# Referências

Aggarwal, C.C. (Ed.), 2014. Data classification: algorithms and applications, Chapman & Hall/CRC data mining and knowledge discovery series. CRC Press/Chapman & Hall, Boca Raton, Fla.

Al, M., 2019. The Best 25 Datasets for Natural Language Processing. lionbridge.ai. URL <https://lionbridge.ai/datasets/the-best-25-datasets-for-natural-language-processing/> (accessed 12.6.19).

ANP, 2019. Instituições credenciadas [WWW Document]. anp.gov.br. URL <http://www.anp.gov.br/pesquisa-desenvolvimento-e-inovacao/credenciamentos-de-instituicoes/instituicoes-credenciadas> (accessed 12.10.19).

Avila, G.V., 2017. Análise de sentimento para textos curtos. FGV EMAp, Rio de Janeiro.

Bansal, S., 2018. A Comprehensive Guide to Understand and Implement Text Classification in Python. Analytics Vidhya. URL <https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/> (accessed 8.14.19).

Brownlee, J., 2017. Deep Learning for Natural Language Processing: Develop Deep Learning Models for your Natural Language Problems. Machine Learning Mastery.

Baeza-Yates, R., Ribeiro-Neto, B., 2013. Recuperação de Informação - 2ed: Conceitos e Tecnologia das Máquinas de Busca. Bookman Editora.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs].

IBICT, 2018. Instituto Brasileiro de Informação em Ciência e Tecnologia [WWW Document]. Instituto Brasileiro de Informação em Ciência e Tecnologia. URL <http://www.ibict.br/sobre-o-ibict/historico>

IBICT, 2019. Indicadores da Biblioteca Digital Brasileira de Teses e Dissertações (BDTD) [WWW Document]. URL <http://bdtd.ibict.br/vufind/Content/statics> (accessed 12.10.19).

International Conference on Machine Learning and Cybernetics, Yeung, D.S., 2006. Advances in machine learning and cybernetics: 4th international conference, ICMLC 2005, Guangzhou, China, August 18-21, 2005 : revised selected papers. Springer, Berlin; New York.

Kim, J.-D., Ohta, T., Tateisi, Y., Tsujii, J., 2003. GENIA corpus--a semantically annotated corpus for bio-textmining. Bioinformatics 19, i180–i182. <https://doi.org/10.1093/bioinformatics/btg1023>

LEXICAL COMPUTING, n.d. Corpus types: monolingual, parallel, multilingual [WWW Document]. sketchengine.eu. URL <https://www.sketchengine.eu/corpora-and-languages/corpus-types/> (accessed 12.5.19).

Makrehchi, M., Kamel, M.S., 2008. Automatic Extraction of Domain-Specific Stopwords from Labeled Documents, in: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (Eds.), Advances in Information Retrieval. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 222–233. <https://doi.org/10.1007/978-3-540-78646-7_22>

Mangiavacchi, J., 2018. Core ML with GloVe Word Embedding and Recursive Neural Network — part 2. Implementing a Natural Language Classifier in iOS with Keras/TensorFlow + Core ML — part 2. Heartbeat. URL <https://heartbeat.fritz.ai/coreml-with-glove-word-embedding-and-recursive-neural-network-part-2-d72c1a66b028> (accessed 12.20.19).

McArthur, T., n.d. The Oxford Companion to the English, 1992nd ed. Oxford & New York: Oxford University Press.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs].

Nguyen, A., 2019. 12 Best Portuguese Language Datasets for Machine Learning. lionbridge.ai. URL <https://lionbridge.ai/datasets/best-portuguese-language-datasets-for-machine-learning/> (accessed 12.7.19).

Obaidullah, S.M., Santosh, K.C., Goncalves, T., Das, N., Roy, K. (Eds.), 2019. Document processing using machine learning, 1st ed. CRC Press, Boca Raton.

Padmanabhan, A., 2019. Text Corpus for NLP. devopedia. URL <https://devopedia.org/text-corpus-for-nlp> (accessed 12.5.19).

Pinto, G.F.S., 2018. Comportamento informacional e mineração textual no Twitter. Universidade Federal de São Carlos.

Ravichandiran, S., 2019. Hands-on deep learning algorithms with python: master deep learning algorithms with extensive math by implementing them using tensorflow.

# Anexos

Notebook dos classificadores

Notebook dos crawlers

- BDTD

- Repositórios

-

Áreas de conhecimento CNPQ

Classificação automática de teses e dissertações

1. Introdução
2. Corpus
   1. O que é um corpus
   2. Para que serve
      1. Vocabulário
      2. Estatística
      3. Modelos vetoriais
         1. Word2Vec
         2. BERT
      4. Modelos *Part-of-Speech* e morfo sintático
   3. Corpus em diversas línguas
      1. Corpus mais famosos
      2. Corpus multilíngues
      3. Corpus em português
   4. Corpus de domínio específico
      1. Corpus gerais e corpus em domínios específicos
      2. Corpus em domínios específicos mais famosos
      3. Corpus em O&G
3. Teses e Dissertações em Óleo e Gás
   1. Vantagens e desvantagens de um corpus formado por teses e dissertações
      1. Vantagens
         1. Público
         2. Termos técnicos e jargões
         3. Documentos relativamente bem padronizados
         4. Documentos longos
      2. Desvantagens
         1. Documentos extraídos de PDF
         2. Jargões acadêmicos x Jargões da indústria
   2. Instituto Brasileiro de Informação em Ciência e Tecnologia - IBICT
   3. Base Digital Brasileira de Teses e Dissertações - BDTD
      1. Estatísticas do BDTD
      2. Instituições presentes no BDTD
      3. Instituições escolhidas
   4. Composição de uma tese e dissertação
      1. Objetivo de um corpus para processamento de linguagem natural
      2. Páginas iniciais e finais
         1. Capa, agradecimentos e sumários
         2. Bibliografia
      3. Corpo do texto
         1. Texto
         2. Figuras
         3. Tabelas
         4. Fórmulas
   5. Metadados das teses e dissertações
4. Classificação
   1. Porque classificar o texto
   2. Engenharia de atributos
      1. *Bag-of-words*
      2. TF-IDF
      3. N-Gram
      4. LDA
      5. Word Embeddings
   3. Algoritmos de classificação
      1. Machine learning clássicos
         1. Naive Bayes
         2. SVM
         3. ...
      2. Deep Learning
         1. CNN
         2. LSTM
         3. ...
5. Metodologia
   1. Visão geral da metodologia
      1. Descrição geral
      2. Figura com *pipeline* da metodologia
   2. *Crawler* na base do BDTD
      1. Quais metadados se queria recuperar
      2. Formatos XML dos repositórios
      3. Tags buscadas
      4. Bibliotecas utilizadas
      5. *Scripts*
      6. Formato final dos metadados extraídos
   3. Classificador de resumos
      1. Base de treino
         1. Uso do *Crawler* BDTD para criar a base de treino
         2. Documentos relevantes para o domínio óleo e gás
            1. Teses e dissertações de empregados da Petrobras
            2. Teses e dissertações com a palavra chave/assunto “Petróleo” e “Gás Natural”
         3. Documentos não relevantes ao domínio óleo e gás
      2. Pré-processamento
         1. O que foi feito
         2. *Scripts*
      3. Engenharia de atributos
         1. O que foi feito
         2. *Scripts*
      4. Treino dos classificadores
         1. O que foi feito
         2. *Scripts*
      5. Métricas e resultados
         1. Descrição das métricas utilizadas
         2. Tabela com os resultados finais
      6. Discussão da escolha do modelo treinado (Deep Learning)
   4. Identificação das teses relevantes para o domínio óleo e gás
      1. Aplicação do classificador já treinado
      2. Extração dos metadados e link para o repositório da instituição de origem
      3. *Scripts*
      4. Estatísticas
   5. *Crawler* nos repositórios das instituições
      1. Recuperação dos documentos
      2. Os diversos tipos de repositórios
      3. Formatos XML dos repositórios
      4. Tags buscadas
      5. Bibliotecas utilizadas
      6. *Scripts*
      7. Estatísticas
   6. Extração da informação dos documentos
      1. ...
      2. ...
6. Corpus final
   1. Estatísticas
7. Conclusão

1. “A body of texts, utterances, or other specimens considered more or less representative of a language, and usually stored as an electronic database. Currently, computer corpora may store many millions of running words, whose features can be analyzed by means of tagging (the addition of identifying and classifying tags to words and other formations) and the use of concordancing programs” [↑](#footnote-ref-1)
2. https://biocreative.bioinformatics.udel.edu/about/background/description/ [↑](#footnote-ref-2)
3. https://academic.oup.com/bioinformatics/article/19/suppl\_1/i180/227927 [↑](#footnote-ref-3)
4. <http://bdtd.ibict.br/vufind/> [↑](#footnote-ref-4)
5. A regulamentação da aplicação dos recursos da cláusula de PD&I podem ser encontrados na Resolução ANP nº 47/2012 e no Regulamento Técnico ANP nº 7/2012. Em: <http://www.anp.gov.br/pesquisa-desenvolvimento-e-inovacao/investimentos-em-p-d-i/regulamentacao-tecnica-relativa-aos-investimentos-em-p-d-i> [↑](#footnote-ref-5)
6. Um programa que automaticamente acessa diversas páginas da internet coletando informações. [↑](#footnote-ref-6)
7. Uniform Resource Locator [↑](#footnote-ref-7)
8. *Hypertext Markup Language* [↑](#footnote-ref-8)
9. Extensible Markup Language [↑](#footnote-ref-9)
10. *JavaScript Object Notation* [↑](#footnote-ref-10)
11. Petróleo Brasileiro S.A. Maior empresa brasileira do setor de petróleo. [↑](#footnote-ref-11)
12. Conselho Nacional de Desenvolvimento Científico e Tecnológico [↑](#footnote-ref-12)
13. https://pypi.org/project/langdetect/ [↑](#footnote-ref-13)
14. https://scikit-learn.org/stable/modules/feature\_extraction.html [↑](#footnote-ref-14)
15. Universidade Federal da Bahia [↑](#footnote-ref-15)
16. <https://tika.apache.org/> [↑](#footnote-ref-16)