**BI Master**

**Trabalho Final – Data Mining**

**Professora Manoela Kohler**

**Alunos:**

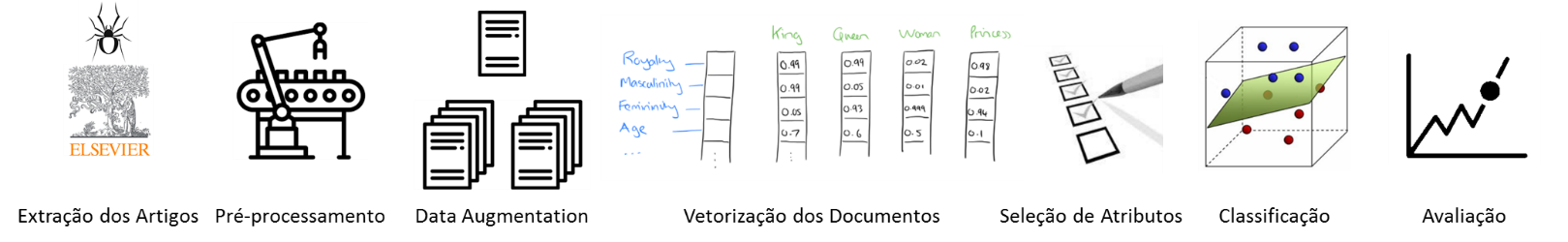
Fábio Corrêa Cordeiro

Gabriel Serrão Seabra

Lívia Fulchignoni de Paiva

**Introdução**

O objetivo do presente trabalho foi treinar um algoritmo de classificação que pudesse identificar se um texto científico foi elaborado por uma empresa do setor de Óleo & Gás. Para tal, foram extraídos cerca de 25.000 resumos de artigos científicos em inglês utilizando a API da Elsevier, uma das maiores editoras científicas do mundo. Com os documentos extraídos foram preparadas as bases de treino e de teste. Foi utilizado um algoritmo de vetorização de texto, Doc2Vec (Le et al., 2014), para se criar os atributos necessários para se treinar um algoritmo de classificação. Os algoritmos de vetorização de textos buscam, de forma não supervisionada, distribuir os textos em um espaço vetorial levando em consideração a semântica dos documentos.

Após testar diversos algoritmos e hiperparâmetros, chegou-se a um F1-score de 86,5% na classificação dos textos utilizando uma *Multilayer Perceptron* (MLP) (Rumelhart et al., 1986) com 5.000 camadas escondidas e vetorização dos textos de dimensão 50.

Atividades realizadas no trabalho

**Extração dos Textos**

A primeira etapa foi a extração dos resumos dos artigos utilizando a API (*Application Program Interface*) da Elsevier[[1]](#footnote-1). Foi necessário realizar duas tarefas, a primeira foi identificar quais os artigos científicos eram relevantes. A segunda tarefa era a extração propriamente dita das informações relevantes e textos que seriam utilizados.

Para a tarefa de identificar quais os artigos eram relevantes foi utilizado o campo “Afiliação” dos autores. Ou seja, caso um dos autores do documento fosse “afiliado” a uma empresa de Petróleo foi considerado que esse era um documento relevante. Esses artigos foram classificados como “O&G” = True. Ao total foram identificados cerca de 15.000 artigos das principais empresas de Petróleo. Os termos utilizados para identificar as empresas de petróleo foram:

* Petrobras
* Exxon
* Shell
* Chevron
* Total
* Eni
* Repsol
* Statoil
* Equinor
* British%20Petroleum[[2]](#footnote-2)
* BP%20International
* BP%20America
* BP%20Exploration
* BP%20United%20Kingdom

Para compor os documentos classificados como “O&G” = False, foram extraídos os primeiros 1.000 documentos de cada ano, iniciando em 2000 e terminando no ano de 2017.

Desta forma, o corpus inicial era composto por 15.000 documentos cujo um dos autores era afiliado a uma empresa de Óleo & Gás e por 17.000 documentos diversos. Cada um desses documentos possuía os seguintes campos:

* Abstract – Texto com alguns parágrafos resumindo o artigo.
* DOI – Código de identificação do documento utilizado pela Elsevier
* Title – Título do documento
* Creator – Lista com o nome dos autores
* Date – Data do documento
* Keywords – Palavras chaves atribuídas para o artigo
* O&G – Identificação de que pelo menos um dos autores era afiliado a uma empresa de petróleo.

O script da extração dos artigos utilizando a API da Elsevier está no arquivo Abstract Extration.ipynb. Já a concatenação de todos os textos extraídos em um único documento está no arquivo Abstract Concatenation.ipynb. O arquivo final, com todos os dados extraídos, está em Elsevier\_abstract – Consolidado.json.

**Pré-processamento**

Antes de iniciar as etapas de vetorização dos textos e de treinar os algoritmos de classificação foi necessário realizar um pré-processamento dos textos. Segue abaixo um exemplo típico de como o texto foi extraído:

*"\n Abstract\n \n Technology advancements and the increasing need for fresh water resources have created the potential for desalination of oil field brine (produced water) to be a cost-effective fresh water resource for beneficial reuse. At the mature oil and gas production areas in the northeast of Brazil, the majority of wells produce a substantial amount of water in comparison with oil production (more than 90%). At these fields, the produced water has to be treated on site only for oil and solids removal aiming re-injection. The aim of this work is to assess the quality of the produced water stream after a reverse osmosis desalination process in terms of physicochemical characteristics influencing reuse of the water for irrigation or other beneficial uses.\n \n "*

Pode-se notar a presença de “\n” que denota quebra de linha, além de vários espaços em branco no início e fim do parágrafo. Também notamos que diversos textos iniciaram com a palavra “Abstract”, ou outra similar *(“Publisher Summary”, "Summary"* ou "Fundamento"). Por fim, alguns textos estavam vazios contendo apenas a palavra “Unknown”.

Para se pré-processar os textos foram executadas as seguintes ações:

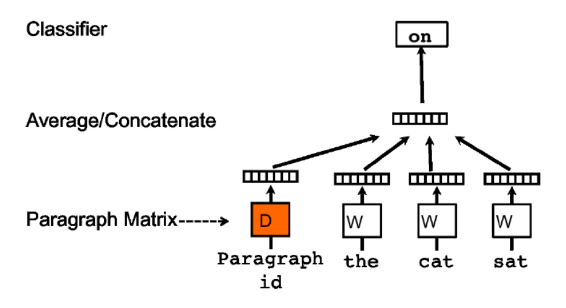
1. Foram excluídas as palavras *“Abstract”, “Publisher Summary”, "Summary"* e "Fundamento".
2. Foi excluída a palavra *“Unknown”.*
3. Foram excluídos os espaços em branco e o símbolo “\n”.
4. Todos os documentos que ficaram sem conteúdo em *“abstract”* foram excluídos.
5. Por fim, foi aplicado o pré-processamento simples da biblioteca Gensim[[3]](#footnote-3). Esse pré-processamento basicamente gera *tokens[[4]](#footnote-4)* a partir dos textos.

Ao final do pré-processamento restaram 25.645 artigos, sendo 14.418 de “O&G” e 11.227 de outras áreas.

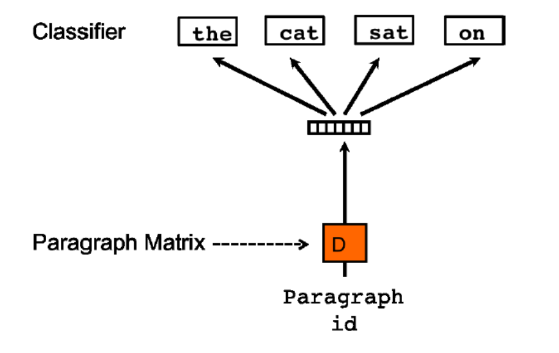
Antes de iniciar as etapas de vetorização dos documentos e treino dos algoritmos de classificação o corpus foi dividido em base de treino e base de teste. A base de treino tinha 90% dos documentos e a de teste 10%. As vetorizações e treinamentos foram realizadas apenas na base de treino.

**Vetorização dos documentos**

O algoritmo de vetorização de documentos utilizado foi o Doc2Vec. Este algoritmo é a implantação dos modelos de “Document Embeddings” proposto por Le et al (2014). Foram propostos dois modelos de vetorização de documentos, um é o Phrase Vector – Distributed Memory” (PV-DM) e o outro é o “Phrase Vector – Distributed bag of words” PV-DBOW. Os dois modelos são estruturas de redes neurais que utilizam vetores de palavras e vetores de documentos para prever as palavras seguintes de uma determinada “janela de palavras”.



Estrutura da rede neural do modelo PV-DM em Le et al. (2014)



Estrutura da rede neural do modelo PV-DBOW em Le et al. (2014)

Foi utilizada a implantação dos algoritmos Doc2Vec disponíveis na biblioteca Gensim[[5]](#footnote-5). Nesta aplicação é possível escolher diversos hiperparâmetros, para esse trabalho variações dos seguintes hiperparâmetros foram testados:

* *Vector size* – Tamanho do vetor que irá representar cada documento
* *DM* - Algoritmo de treino a ser usado, PV-DM ou PV-DBOW
* *Window* – Distância máxima entre a palavra atual e a palavra a ser predita
* *Epochs* – Número de iterações sobre o corpus
* *Min Count* – Frequência mínima da palavra no corpus para ser considerada

Inicialmente, os vetores são iniciados com um valor aleatório para cada documento. Conforme o treinamento vai ocorrendo ao longo dos diversos documentos e das diversas épocas de iteração, os valores dos vetores vão se aproximando da representação semântica do conteúdo do documento.

Ao final da fase de treinamento, o modelo treinado consegue inferir um vetor para um determinado texto que lhe é informado. A fase seguinte, consequentemente, consiste em inferir um vetor para todos os artigos, tanto para a base de treino quanto para a base de teste.

**Classificação**

Para a classificação foram testadas duas estratégias diferentes. A primeira consistia em utilizar os vetores do algoritmo Doc2Vec como atributos para os algoritmos de classificação e a coluna “O&G” como classe a ser prevista. Em seguida utilizou alguns dos algoritmos da biblioteca Scikit Learn[[6]](#footnote-6) para treinar e avaliar o classificador.

A segunda abordagem iniciava treinando um algoritmo *“Decision tree”* exatamente da mesma forma que na primeira abordagem. Em seguida, utilizou-se o atributo “*feature\_importances\_”* do algoritmo *“Decision tree”* para se identificar quais os atributos do vetor Doc2Vec eram mais importantes para se prever “O&G”. Dessa forma, foi possível utilizar no classificador um número menor de atributos do que o tamanho do vetor treinado. Essa abordagem tentou representar os documentos em um espaço vetorial mais amplo, permitindo uma representação semântica mais variada, e ao mesmo evitar *“overfiting”* dos modelos de classificação.

Os modelos de classificação testado[[7]](#footnote-7), sempre usando biblioteca Scikit Learn, foram:

* *Support Vector Machine*
* *Decision Tree*
* *Random Forest* com 100 estimadores
* *Near Neighbors*
* *Near Centroid*
* *Gaussian Naive Bayes*
* *Multilayer Perceptron* com solver *“Stochastic gradiente descente”*
* *Multilayer Perceptron* com solver *“Stochastic gradiente descente” e* 5.000 camadas escondidas

As métricas de avaliação utilizadas para a escolha dos hiperparâmetros e algoritmos foi F1-score. Também foi calculado os valores de acurácia, precisão e revocação.

***Data augmentation***

Uma técnica testada foi o “*data augmentation”*. O objetivo era ampliar o conjunto de treino possibilitando um melhor treinamento dos algoritmos de vetorização e de classificação. Também foi aplicado um certo desbalanceamento de forma que os documentos “O&G” ficassem sobrerrepresentados. O intuito era de representar com mais detalhes os atributos relacionados aos documentos de “O&G”.

Após o “*data augmentation”* os documentos referentes à “O&G” foram triplicados e os demais foram duplicados. Por fim, o total de documentos treinados era aproximadamente 60.000, sendo 2/3 de “O&G”.

**Seleção de hiperparâmetros e algoritmos de classificação**

Para se iniciar os testes dos diversos hiperparâmetros e algoritmos possíveis foi definido um caso base para em seguida ir variando os hiperparâmetros um a um. O Caso Base era o seguinte:

Doc2Vec

* Vector Size – 100
* Algoritmo – PV-DBOW
* Min Count – 1
* Epoch – 100
* Window – 5

Algoritmo de Classificação – *Support Vector Machine*

Teste/ Treino – 10% / 90%

Data Augmentation – Não

Número de atributos usados para treino – 100

Após treinar e classificar usando os hiperparâmetros do caso base chegou-se às seguintes métricas de classificação:

**F1- score:** 85,06%

**Acurácia:** 85,11%

**Precisão:** 85,09%

**Revocação:** 85,11%

Ao variar os hiperparâmetros um a um, mantendo os demais constantes, identificamos aqueles que melhoravam as métricas de performance. São eles:

Classificador – Multilayer Perceptron com 5000 camadas escondidas

Vector Size – 20

Com esses novos parâmetros criamos um Caso Base v2 com as seguintes métricas de classificação:

**F1- score:** 85,69%

**Acurácia:** 85,73%

**Precisão:** 85,77%

**Revocação:** 85,73%

Finalmente, foram feitos diversos outros testes variando mais de um atributo no entorno do Caso Base v2 e dos demais casos que apresentaram boas métricas de classificação. O Caso Final que obteve as melhores métricas foi:

Doc2Vec

* Vector Size – 50
* Algoritmo – PV-DBOW
* Min Count – 1
* Epoch – 100
* Window – 5

Algoritmo de Classificação – Multilayer Perceptron com 5000 camadas escondidas

Teste/ Treino – 10% / 90%

Data Augmentation – Não

Número de atributos usados para treino – 50

**F1- score:** 86,55%

**Acurácia:** 86,59%

**Precisão:** 86,63%

**Revocação:** 86,59%



Testes de seleção de hiperparâmetros e algoritmos de classificação

**Conclusão**

A vetorização de texto pode ser utilizada em combinação com os algoritmos clássicos em tarefas de classificação. No entanto, apesar de se possuir um corpus relativamente grande e testando diversas combinações de hiperparâmtros, essa técnica demonstrou não ultrapassar a barreira de F1-score de 87%. Para muitas tarefas esse resultado é “boa o suficiente” e com certeza realiza a tarefa muito mais rápido do que seres humanos. No entanto, se for necessário uma acurácia maior será preciso utilizar uma técnica mais robustas e com custo computacional maior, como as técnicas de Deep Learning para Processamento de Linguagem Natural.[[8]](#footnote-8)

**Bibliografia**

Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprinted arXiv:1810.04805v1, 2018. Em: https://arxiv.org/pdf/1810.04805.pdf

Le, Quoc; Mikolov, Tomas. Distributed Representations of Sentences and Documents. arXiv preprinted arXiv:1405.4053v2, 2014. Em: <https://arxiv.org/pdf/1405.4053v2.pdf>

Rumelhart, David; Hinton, Geoffrey; Williams, Ronald. Learning representations by back-propagatings errors. Nature Vol 323, 1986. Em: <https://www.iro.umontreal.ca/~pift6266/A06/refs/backprop_old.pdf>

1. As instruções de como se utilizar a API pode ser encontrada na seção “How to Guides -> Text Mining” em: https://dev.elsevier.com/tecdoc\_text\_mining.html . [↑](#footnote-ref-1)
2. O termo BP retornou diversos documentos que não estavam relacionados à petroleira Britânica, portanto foram utilizado diversos termos para tentar capturar os artigos dessa empresa. [↑](#footnote-ref-2)
3. Biblioteca em python para modelagem semântica de textos. [↑](#footnote-ref-3)
4. Tokem é a unidade mínima utilizado nas técnicas de processamento de linguagem natural. Em geral é uma sequência de texto entre dois espaços ou entre espaços e símbolos de pontuação. [↑](#footnote-ref-4)
5. https://radimrehurek.com/gensim/models/doc2vec.html [↑](#footnote-ref-5)
6. https://scikit-learn.org/stable/supervised\_learning.html#supervised-learning [↑](#footnote-ref-6)
7. Quando não informado, os hiperparâmetro utilizados foram os padrões sugeridos pela biblioteca Scikit Learn. [↑](#footnote-ref-7)
8. Há poucos meses foi o Google AI Language publicou o último benchmark nas tarefas de processamento de linguagem natural utilizando *deep learning* (Devlin et al. 2018)*.* [↑](#footnote-ref-8)