Categorização automática de conjuntos de dados de portais de dados abertos utilizando aprendizado de máquina supervisionado

Automatic datasets categorization of open data portals using supervised machine learning

Mateus de Moraes Rangel[1]

Resumo

Para disponibilizar seus dados para a sociedade, governos de cidades ao redor do mundo estão usando portais de dados abertos. Na maioria dos portais, os conjuntos de dados estão distribuídos por categorias que representam os tópicos abordados pelo portal. Nesse contexto, oferecer mecanismos para auxiliar a categorização dos conjuntos de dados se torna importante, para facilitar o trabalho de um administrador de portais de dados abertos. Neste trabalho, apresentamos uma metodologia para a categorização automática de conjuntos de dados de portais de dados abertos. Em nossa metodologia, utilizamos o nome do conjunto de dados e os seus atributos de arquivos anexados para a inferência de sua categoria, fazendo uso de técnicas de processamento de linguagem natural e aprendizado de máquina supervisionado.

Palavras-chave: Processamento de linguagem natural. Aprendizado de máquina supervisionado. Dados abertos.

Abstract

Trabalho de conclusão de curso apresentado ao curso de Bacharelado em Sistemas de Informação da Universidade Federal Fluminense como requisito parcial para conclusão do curso.

[1] Graduando do Curso de Sistemas de Informação-UFF; mateusrangel@id.uff.br.

To make their data available to society, city governments around the world are using open data portals. In most portals, datasets are broken down into categories that represent the topics covered by the portal. In this context, providing mechanisms to help categorize datasets becomes important to facilitate the work of an open data portal administrator. In this paper, we present a methodology for the automatic categorization of data sets from open data portals. In our methodology, we use the dataset name and its attached file attributes to infer its category, making use of natural language processing techniques and supervised machine learning.

Keywords: Natural language processing. Supervised machine learning. Open Data.

Aprovado em: 20/12/2019 Versão Final em: 20/12/2019

1 Introdução

Nos últimos anos, governos de cidades ao redor do mundo vêm disponibilizando seus dados de forma aberta por meio de portais na internet, como uma forma de atender à demanda de transparência da sociedade. Por meio desses portais, a sociedade pode consultar e requisitar bases de dados para obter informações úteis sobre áreas como saúde, transporte, segurança, etc.

Um dos desafios que são encontrados ao fazer alguma análise sobre portais de um conjunto de cidades é a sua integração do ponto de vista das categorias em que os conjuntos de dados estão agrupados. Por não haver um padrão pré-definido, existem casos em que portais se referem à mesma área porém com nomes de categorias diferentes, como “segurança"e “segurança pública". Uma maneira para facilitar a integração de conjuntos de dados, proposta em (PINTO; BERNARDINI; VITERBO, 2018), consiste na geração de um subconjunto abrangente de categorias a partir do conjunto de categorias dos portais que se deseja integrar. Nesse contexto, oferecer mecanismos para auxiliar a categorização é importante, para facilitar o trabalho de um administrador do portal de dados governamentais abertos.

1.1 Objetivo e Metodologia da Pesquisa

O objetivo deste trabalho é apresentar uma metodologia para, dado um conjunto de categorias, categorizar de maneira automática um conjunto de dados a ser disponibilizado em um portal de dados governamentais abertos baseado em aprendizado de máquina. Para isso os seguintes passos foram executados:

- Estudo da fundamentação teórica para este trabalho, que envolveu: (i) Estudo de técnicas de Processamento de Linguagem Natural (PLN); (ii) Estudo de algoritmos de aprendizado de máquina; e (iii) Estudo do estado da arte em integração de dados em portais de dados governamentais abertos.
- Proposta de metodologia para categorização automática dos conjuntos de dados.

- Avaliação da metodologia proposta utilizando dados coletados de portais de dados governamentais abertos, incluindo conjunto de categorias e metadados dos conjuntos de dados.

1.2 Organização do trabalho

Este trabalho está dividido como segue: No Capítulo 2 é apresentada a fundamentação teórica e revisão da literatura para este trabalho. No Capítulo 3 é apresentada a metodologia proposta. No Capítulo 4 é apresentada a avaliação da metodologia proposta e são apresentados os resultados obtidos nessa avaliação. No Capítulo 5 são apresentadas as conclusões e as sugestões para trabalhos futuros.

2 Fundamentação Teórica

2.1 Dados Abertos

A Open Knowledge International (INTERNATIONAL, b) define dados abertos como: "dados e conteúdos que podem ser usados, modificados e compartilhados por qualquer pessoa para qualquer propósito" (INTERNATIONAL, a). Suas principais características são (INTERNATIONAL, c):

- Disponibilidade: os dados devem estar disponíveis a um custo de reprodução razoável, de preferência através de *download* pela internet. Os dados também devem estar disponíveis de forma conveniente e modificável.
- Reutilização e redistribuição: os dados devem ser fornecidos em termos que permitam a reutilização e redistribuição, incluindo o intercâmbio com outros conjuntos de dados. Os dados devem ser legíveis por máquina.
- Participação universal: todos devem poder usar, reutilizar e redistribuir, não deve haver discriminação contra os campos de trabalho ou contra pessoas ou grupos.

2.2 Portais de Dados Governamentais Abertos

Para tornar seus dados públicos e, consequentemente, promover a transparência, governos de cidades ao redor do mundo fazem uso de portais de dados. Esses portais são formados por páginas *web* onde conjuntos de dados sobre várias áreas de atuação do governo da cidade podem ser publicados seguindo a definição de dados abertos descrita anteriormente.

As categorias disponíveis em um portal de dados governamentais abertos representam os assuntos que foram abordados em seus conjuntos de dados, como serviços, transporte, planejamento, finanças e saúde. Informações como nome, descrição, licença, mantenedor, data de criação, data de modificação, autor, entre outras, costumam ser encontradas nos arquivos de metadados dos conjuntos de dados de portais de dados abertos.(BARBOSA et al., 2014)

2.3 Processamento de Linguagem Natural

Processamento de Linguagem Natural (PLN) é uma área de estudo voltada para a extração de significado/padrões de texto em idiomas humanos. Muitas técnicas de PLN usam aprendizado de máquina para derivar significado do corpus textual em questão. Suas áreas de aplicação podem incluir: categorização de textos (como neste trabalho), reconhecimento de entidade nomeada e geração de textos. (MANNING; SCHüTZE, 1999)

2.3.1 Técnicas de vetorização textual

Algoritmos de aprendizado de máquina em geral aceitam como entrada dados em formato atributo-valor. Para que possamos aplicar algoritmos de aprendizado de máquina em textos, é necessário transformar o conteúdo de textos nesse formato, o que implica que cada documento passa a ser representado como um vetor numérico. A seguir, apresentamos as técnicas utilizadas neste trabalho.
(MANNING; RAGHAVAN; SCHüTZE, 2008)

Bag of Words

Na abordagem de vetorização *bag-of-words* (saco de palavras), um vocabulário contendo todas as palavras que estiveram presentes em pelo menos um dos exemplos do conjunto de dados é gerado. Em seguida, cada exemplo do conjunto de dados é representado como um vetor que indica a quantidade de vezes que a palavra esteve presente no mesmo. Uma *bag-of-words* é dita binária se representar apenas a presença da palavra em um exemplo.
(MANNING; RAGHAVAN; SCHüTZE, 2008)

***Term Frequency–Inverse Document Frequency*(TF-IDF)**

Em problemas de categorização de textos, tipicamente as categorias são identificadas por palavras que as caracterizam. À primeira vista podemos pensar que as palavras que mais aparecem em documentos pertencentes à uma categoria são as que a definem. Porém, os melhores indicadores de uma categoria são as palavras que mais aparecem naquela categoria e não em outras.(RAJARAMAN; ULLMAN, 2011)

A medida formal de o quão concentrada em relativamente menos documentos são as ocorrências de uma palavra é chamada TF.IDF (RAJARAMAN; ULLMAN, 2011) sendo uma abreviação do inglês para *Term Frequency times Inverse Document Frequency* que significa frequência do termo multiplicado pelo inverso da frequência nos documentos.

Suponha que temos uma coleção de N documentos. Defina f_{ij} a frequência do termo i no documento j. Então, definimos *term frequency* pela Eq. 1: em que a frequência do termo i no documento j (f_{ij}) é normalizada pela divisão da maior ocorrência de qualquer termo k no mesmo documento($max_k f_{kj}$).

$$TF_{ij} = \frac{f_{ij}}{max_k f_{kj}} \tag{1}$$

Já o IDF de um termo é definido pela Equação 2 onde N é o número de documentos

e n_i é o número de documentos em que o termo i está presente.

$$IDF_i = log_2 \left(\frac{N}{n_i} \right) \tag{2}$$

Com isso podemos calcular o TF.IDF de um termo i em um documento j como TF_{ij} x IDF_i. Os termos com maior TF.IDF *score* são geralmente os termos que melhor caracterizam o tópico do documento.

2.3.2 *Stopwords*

No contexto de processamento de linguagem natural, as *stopwords* (palavras vazias) são um conjunto de palavras, geralmente as mais comuns em um texto em um idioma(como "a", "uma", "isso" no caso do português). (MANNING; RAGHAVAN; SCHüTZE, 2008)

Em um tarefa de classificação textual, as *stopwords* são removidas para que haja um foco maior nas palavras que definem o significado do texto e sua classe, podendo, assim, aumentar a acurácia de classificação. Neste trabalho, por termos utilizado como fonte de dados portais de dados abertos americanos, o conjunto de *stopwords* utilizado foram as *stopwords* do idioma inglês.

2.3.3 *Stemming*

Em PLN, *stemming* é um processo para remover as terminações morfológicas e inflexionais mais comuns das palavras. Seu principal uso é na parte de normalização de termos.(PORTER, 2006)

Neste trabalho, o algoritmo utilizado foi o Porter Stemmer na implementação do NLTK. O algoritmo Porter Stemmer tem esse nome por ter sido publicado Martin Porter em (M.F.PORTER, 1980). Um conjunto de implementações, dúvidas e referências podem ser encontradas na página *web* (PORTER, 2006) mantida pelo próprio autor.

2.3.4 NLTK - *Natural Language Toolkit*

O NLTK(NLTK, 2019) é uma plataforma para desenvolvimento de programas feita na linguagem de programação Python para tarefas de PLN. Ela disponibiliza um pacote de bibliotecas de processamento textual para classificação, tokenização, *stemming*, etc. Neste trabalho utilizamos a sua implementação do porter stemmer e seu conjunto de *stopwords* em inglês.

2.4 Aprendizado de máquina

Aprendizado de máquina (FACELI et al., 2011) é uma área de estudo voltada para que computadores possam reconhecer padrões após aprender com experiência passada. Para isso, é empregado um princípio de inferência, denominado indução, no qual se obtém conclusões genéricas a partir de um conjunto de exemplos.

Os dois principais tipos de tarefa de aprendizado de máquina são: aprendizado supervisionado e aprendizado não supervisionado. Em tarefas supervisionadas, queremos gerar uma função a partir dos dados de treinamento que possa ser utilizada para prever um

rótulo(classificação) ou valor(regressão) que caracterize um novo exemplo cujo atributo de saída é desconhecido. Em tarefas não supervisionadas, queremos descrever um conjunto de dados. Diferente das tarefas de previsão, não há atributo de saída. Formas de explorar os dados são: agrupamento, associação e sumarização.(FACELI et al., 2011)

Nesta seção, são apresentados os algoritmos de aprendizado bayesiano, a ferramenta scikit-learn, que implementa os algoritmos apresentados na linguagem Python, e as medidas de avaliação, todos utilizados neste trabalho.

2.4.1 Algoritmos de Aprendizado Bayesiano

O teorema de bayes fornece uma maneira de calcular a probabilidade de um evento ou objeto B pertencer a uma classe A, $P(A|B)$, utilizando a probabilidade a priori da classe, $P(A)$, a probabilidade de obserar vários objetos que pertencem à classe, $P(B|A)$, e a probabiliade de ocorrência desses objetos, $P(B)$, e pode ser expresso pela Equação. 3 (FACELI et al., 2011)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3}$$

Naive Bayes

O classificador Naive Bayes(Bayes Ingênuo) possui esse nome por assumir que os valores dos atributos de um exemplo são independentes entre si dada a classe, $P(x|y_i)$ pode ser decomposto no produto $P(x^1|y_i) \times ... \times P(x^d|y_i)$, em que x^j é o j-ésimo atributo do exemplo x. Com isso, a probabilidade de um exemplo x pertencer à classe y_i é proporcional à expressão:

$$P(y_i|x) \propto P(y_i) \prod_{j=1}^{d} P(x^j|y_i) \tag{4}$$

Suponha que queremos classificar um exemplo x entre duas ou mais classes. Usando o método de estimativa(regra de decisão) MAP (*Maximum A Posteriori*), considerando cada classe y_i, classificamos o exemplo x como sendo da classe que obtiver a maior probabilidade segundo a Eq. 5, ou seja, escolhemos a hipotese mais provavel.

$$y_{MAP} = \underset{i}{\operatorname{argmax}}\, P(y_i|x) \tag{5}$$

2.4.2 Multinomial Naive Bayes

Multinomial Naive Bayes (MANNING; RAGHAVAN; SCHüTZE, 2008) se trata de um *event model* do classificador naive bayes. *Event model* são suposições sobre a distribuição dos atributos. Utilizando o multinomial naive bayes, a probabilidade do j-ésimo atributo do exemplo x pertencer à classe y_i, $P(x^j|y_i)$, na expressão 4, é calculada como a frequência relativa do termo j em documentos pertencentes à classe y_i e é dada pela equação 6. Onde T_{ct} é o número de ocorrências do termo t em documentos de treino da classe c, e $\sum_{t' \in V} T_{ct'}$ é a contagem do número de ocorrências de todos os termos do vocabulário V na classe c. Repare que estamos adicionando 1 para cada contagem

para eliminar zeros. Isso é feito para podermos tratar casos em que estamos calculando a probabilidade de um termo que não estava em nenhum exemplo do conjunto de treinamento, essa técnica se chama *Laplace smoothing*.

$$P(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} \tag{6}$$

2.4.3 Máquinas de Vetor Suporte

As Máquinas de Vetor Suporte (ou *Support Vector Machine - SVM*) são modelos utilizados para classificação ou regressão baseados em vetores de suporte. Assim, solução para o problema de construção da SVM é dependente apenas de um subconjunto de exemplos de treinamento, que são os vetores de suporte. Para sua construção, são utilizados algoritmos de aprendizado baseados na teoria de aprendizado estatístico. Inicialmente foram propostos para problemas de classificação binária linear, mas posteriormente o conceito no qual se baseiam esses algoritmos foi ampliado para ser utilizado em problemas de classificação não linear, agrupamento de dados (aprendizagem não-supervisionada), classificação multiclasse e regressão. Esses algoritmos resolvem um problema de otimização quadrática, para minimização de uma função lagrangiana, cuja solução possui ampla e estabelecida teoria matemática. A complexidade de tempo de treinamento da maioria dos algoritmos para construção de SVMs é quadrática ou cúbica em relação ao número de amostras de treinamento, o que é uma das desvantagens desses algoritmos. Ainda, são diversos os parâmetros que podem ser utilizados para sua construção, o que também demanda tempo de experimentação para cada domínio de aplicação. Ainda assim, têm sido bastante exploradas pois em muitos problemas no aprendizado de máquina as SVMs têm apresentado bons resultados (FACELI et al., 2011). Deve ser observado que neste trabalho foram utilizados algoritmos para construção de máquinas de vetor suporte lineares.

2.4.4 Medidas de desempenho

As medidas utilizadas no cálculo do desempenho do classificador em nosso trabalho são:

- Acurácia (Acc): proporção de exemplos corretamente classificados.
- Precisão ($Prec$): proporção de exemplos de uma classe C classificados corretamente entre todos aqueles preditos como pertencentes à classe C.
- Revocação ($Recall$): proporção de exemplos da classe C que foram corretamente preditos.
- F1-Score: Como a precisão não diz quantos exemplos da classe C não foram classificados corretamente e a revocação não diz quantos outros exemplos foram classificados incorretamente como pertencendo à classe C. A precisão e a revocação

são combinadas na medida F1- score que consiste na média harmônica da precisão ($prec$) e a revocação ($recall$) e é apresentada na Eq. 7 :

$$F1 = \frac{2 \times Prec \times Recall}{Prec + Recall} \tag{7}$$

Por estarmos em um problema multiclasse, a precisão, revocação e f1-score foram calculadas usando a média macro. Na média macro, a métrica é calculada para cada classe, e em seguida é calculada a sua média aritmética sem peso por classe.

2.4.5 scikit-learn

Scikit-learn (LEARN, 2019) é uma biblioteca de aprendizado de máquina para a linguagem de programação Python. Ela inclui, além de algoritmos clássicos de aprendizado de máquina, métodos para todo ciclo de vida do processo de aprendizado de máquina como pré-processamento, redução de dimensionalidade, avaliação/seleção de modelos, etc.

2.5 Revisão da Literatura

Pinto, Bernardini e Viterbo (PINTO; BERNARDINI; VITERBO, 2018) apresentaram uma pesquisa utilizando como fonte de dados 100 portais de cidades americanas mais densamente populosas. Nesta pesquisa, mostram como os portais categorizam seus conjunto de dados para que possam obter as categorias mais significativas. E sugerem um método para prover uma categorização genérica para conjuntos de dados pertencentes a portais de dados governamentais abertos.

Com base no trabalho anterior, (PINTO, 2018) apresenta com mais detalhe o processo de obtenção do Subconjunto Categorias, denominado Subconjunto Abrangente, disponibilizando o código fonte do algoritmo(que utilizamos neste trabalho). Abrangente neste contexto, significa que o conjunto de categorias gerado consegue descrever grande parte dos dados de todos os portais utilizados como entrada.

Ainda no mesmo trabalho, apresentam um processo de alinhamento de categorias de portais com o subconjunto abrangente de categorias. Neste processo, cada categoria de cada portal é alinhada com uma das categorias mais abrangentes previamente calculadas utilizando o cálculo de similaridade semântica(MIHALCEA; CORLEY; STRAPPARAVA, 2006). O cálculo da similaridade semântica entre uma categoria de um portal e uma categoria do subconjunto abrangente é feito através do cálculo da similaridade semântica entre todas as palavras que formam as categorias.

Como forma de categorizar conjuntos de dados ainda sem categoria em um portal de dados abertos, (Frtunić Gligorijević et al., 2019) introduzem um classificador chamado EODClassifier framework. Este classificador tem como base a análise formal do conceito como forma de gerar uma estrutura de dados que revela uma conceitualização compartilhada originada do uso de tags.

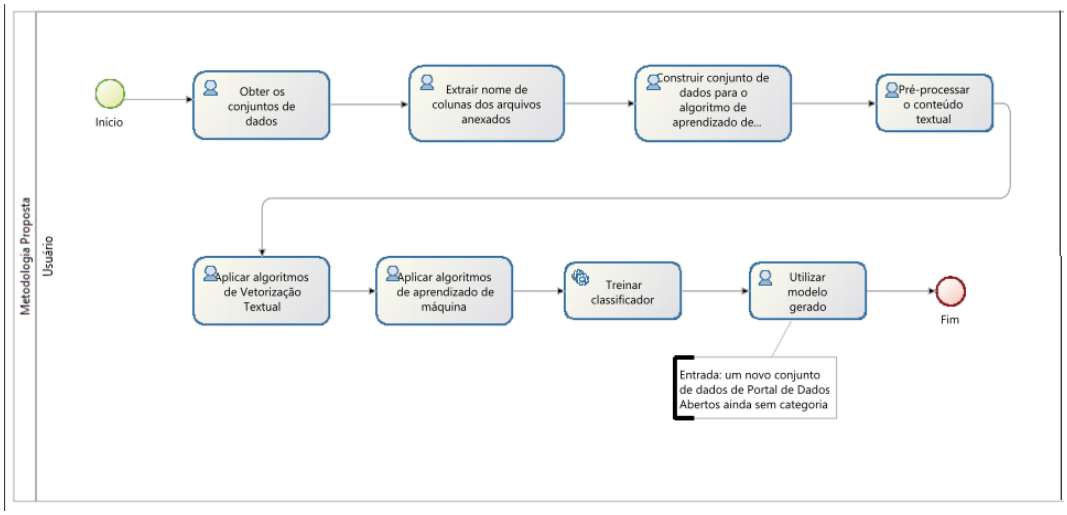
Pelo fato de portais de dados governamentais abertos não seguirem um padrão de estruturas de categorização, (YANG; LIN; YU, 2015) tentam avaliar a qualidade da

estrutura de categorização de portais de dados abertos automaticamente ao investigar a similaridade dos conjuntos de dados que estão contidos na mesma categoria.

3 Metodologia

Na Figura 1 é apresentado o processo de execução da metodologia proposta neste trabalho. Foi utilizada a ferramenta Bonita Studio 7.9.4 para desenho do processo. Cada passo do processo é descrito nas subseções a seguir.

Figura 1 – Processo da metodologia proposta.



3.1 Obter os conjuntos de dados

Na nossa metodologia, o termo usuário se refere aos administradores de portais de dados abertos. O usuário deve ter acesso aos conjuntos de dados do portal de dados abertos desejado contendo: Seu nome, arquivos anexados e categoria. A coleta dos dados pode ser realizada manualmente ou automaticamente.

3.2 Extrair nome de colunas dos arquivos anexados

Para cada tipo de arquivo anexado que um conjunto de dados tiver, devem ser obtidos os nomes de suas colunas. A obtenção das colunas vai depender do tipo de arquivo e sua respectiva formatação. Tipos de arquivos bastante comuns em arquivos anexados de portais de dados abertos são CSV(*Comma-separated values*) e GeoJSON. Em arquivos CSV, seus nomes de coluna normalmente se encontram na primeira linha do arquivo sendo separados por vírgula. Em arquivos GeoJSON, seus nomes de coluna são encontrados dentro do objeto FeatureCollection, dentro de objetos contidos no *array features*, sendo as chaves dentro do objeto *properties*.

3.3 Construir conjunto de dados para o algoritmo de aprendizado de máquina

Para cada conjunto de dados coletado, deve ser criada uma tupla contendo os seguintes campos: texto e categoria. Onde texto é a concatenação do nome do conjunto de dados com suas colunas extraídas sendo unidos por espaço; e categoria é a categoria do conjunto de dados.

3.4 Pré-Processar o conteúdo textual

Para que haja um melhor aproveitamento do conteúdo textual do nome e colunas do conjunto de dados, as seguintes etapas de pré-processamento textual devem ser seguidas:

- SEPARAR POR *UNDERLINE*: Um dos padrões na nomenclatura de colunas/atributos em bases de dados é a separação de palavras por *underline*. Para separar uma sentença unida por *underline*, basta substituir o *underline* com um espaço em branco.
- SEPARAR POR *CAMEL CASE*: Um dos padrões na nomenclatura de colunas em arquivos de dados é a união de palavras por *camel case*. Para separar uma sentença unida por *camel case*, ao detectar a mudança de *casing* em um nome de uma coluna, considere todos os caracteres que vieram antes da mudança como um termo e o restante como outro termo e os una com espaço.
- RETIRAR OS NÚMEROS: Números no nome e colunas de um conjunto de dados não costumam adicionar nenhum valor no contexto de uma tarefa de classificação textual. Durante as primeiras iterações do processo de avaliação da metodologia, detectamos muitos anos como *tokens* em nossos vetores textuais, por isso fizemos a remoção.
- RETIRAR AS *STOP WORDS*: Além de tirar as palavras de cada exemplo que estiverem contidas no conjunto de *stopwords* do idioma em que o conjunto de dados é escrito, criamos uma lista de palavras que são muito comuns em arquivos de dados abertos e georreferenciados, sendo elas: 'objectid', 'shape', 'length', 'area', 'y', 'x', 'id', 'zip', 'date', 'address', 'code', 'street', 'district', 'lat', 'lng', 'latitude', 'longitude'. Esse conjunto de termos foi obtido após detectarmos em nossas primeiras iterações de avaliação da metodologia a grande frequência que esses termos apareciam em nossos vetores textuais.
- *STEMMING*: Após as modificações anteriores, os termos são transformados para seus radicais. Palavras como "station"e "stations"podem ser tratadas como um só termo ("statio", por exemplo) e ajudam no passo posterior de vetorização textual. Para a língua inglesa, o algoritmo porter stemmer é comumente utilizado.

3.5 Aplicar algoritmos de Vetorização Textual

Com todo o conteúdo textual do conjunto de dados pré-processado, técnicas de vetorização textual já podem ser aplicadas para que algoritmos de aprendizado de máquina possam ser treinados. Técnicas muito comuns envolvem o uso de construção de *bag-of-words* e TF-IDF, apresentadas anteriormente.

3.6 Aplicar algoritmos de aprendizado de máquina

Com o *dataset* já transformado para o formato de vetores numéricos, o usuário já pode aplicar algoritmos de aprendizado de máquina para gerar um classificador. Algoritmos comumente utilizados para essa tarefa são: Naive Bayes e Support Vector Machines.

3.7 Utilizar modelo gerado

Após ter treinado o classificador escolhido, o usuário pode obter uma predição automática de um conjunto de dados ainda sem categoria. Para isso, devem ser seguidos os passos apresentados nas Seções 3.2, 3.3, 3.4 e 3.5 para obter a representação vetorial do conjunto de dados e então dar como entrada para o classificador gerado na Seção 3.6.

4 Avaliação da Metodologia

Por estarmos em uma tarefa de aprendizado de máquina supervisionado, podemos analisar o desempenho do nosso modelo ao testá-lo com novos exemplos onde já sabemos sua categoria, porém não os apresentamos em sua fase de treinamento. Para isso, foi utilizada a técnica de amostragem *holdout* (FACELI et al., 2011). No *holdout*, o conjunto de dados gerado na etapa de construção de *dataset* é aleatoriamente dividido em conjunto de treinamento e teste. Em nosso caso, a proporção escolhida foi 2/3 para treinamento e 1/3 para teste. A seguir, falaremos o que foi feito em cada passo da metodologia apresentada anteriormente.

4.1 Obter conjuntos de dados

Para a construção de nosso conjunto de dados, foram utilizados 5 portais de cidades americanas dentre as 100 cidades americanas mais populosas abordadas por (PINTO; BERNARDINI; VITERBO, 2018), que usam o CKAN como plataforma de portais de dados abertos: Houston [2], Philadelphia [3], Lexington [4], Newark[5] e Birgmingham[6].

Para que os portais possuam o mesmo conjunto de categorias, executamos a rotina de geração de subconjunto abrangente de categorias dos portais desenvolvida e disponibilizada em (PINTO, 2018). Após a execução, 8 categorias foram geradas, sendo elas: *Services*, *Transportation*, *Planning*, *Development*, *Safety*, *Finance*, *Health*, *Engineering*.

Neste trabalho, usamos apenas os conjuntos de dados cujas categorias tenham sido usadas para formar o subconjunto de categorias abrangente. Ou seja, categorias cujo nome possua uma das palavras mais abrangentes, exemplo: a categoria 'Public Works & Engineering' faz parte do conjunto de categorias que possui a palavra 'engineering' que foi usada para obter a categoria abrangente 'engineering'. A distribuição de frequências das categorias dos conjuntos de dados utilizados está apresentada na Tabela 1.

[2] Disponível em http://data.houstontx.gov/
[3] Disponível em https://www.opendataphilly.org/
[4] Disponível em https://data.lexingtonky.gov/
[5] Disponível em http://data.ci.newark.nj.us/
[6] Disponível em https://data.birminghamal.gov/

Tabela 1 – Frequência absoluta e relativa das categorias dos conjuntos de dados

Categoria	Frequência Absoluta	Frequência Relativa
Services	56	0.254545
Transportation	49	0.222727
Planning	43	0.195455
Development	32	0.145455
Safety	18	0.081818
Finance	11	0.050000
Health	7	0.031818
Engineering	4	0.018182

4.2 Construir conjunto de dados para o algoritmo de aprendizado de máquina

A construção do conjunto de dados em formato atributo valor para a execução dos algoritmos de aprendizado de máquina se deu da seguinte forma: Para cada conjunto de dados de cada portal, que é um exemplo de treinamento e teste para o aprendizado de máquina, criamos uma tupla com os atributos texto e *category*. O atributo texto contém o nome do conjunto de dados e o nome dos atributos de seus arquivos anexados e o atributo *category* contém a categoria do conjunto de dados. Apenas arquivos na forma CSV e GeoJSON tiveram seus atributos extraídos. Está disponível em https://github.com/mateusrangel/tcc-resources/blob/master/corpus.csv o arquivo contendo cada exemplo (dados extraídos de cada conjunto de dados de cada portal) com sua respectiva categoria.

4.3 Pré-Processar o conteúdo textual

Na etapa de pré processamento, além das etapas de tratamento textual, descritas na Seção 3.4, também foi feita a remoção de exemplos que sejam da categoria *Health* ou *Engineering*, por terem poucos exemplos no conjunto de dados gerado, como mostrado na Tabela 1.

4.4 Aplicar algoritmos de Vetorização Textual

Para que possamos medir o desempenho de nossa metodologia em diferentes técnicas de vetorização textual, foram utilizadas três técnicas já apresentadas anteriormente: *bag-of-words*, *bag-of-words* binária e TF-IDF.

4.5 Aplicar algoritmos de aprendizado de máquina

Neste trabalho, usamos os algoritmos Multinomial Naive Bayes e Máquinas de Vetores de Suporte Lineares com os parâmetros padrões da biblioteca scikit-learn.

4.6 Resultados

A seguir apresentamos o desempenho de nossa metodologia com base nos valores das principais métricas de uma tarefa de classificação em aprendizado de máquina,

Tabela 2 – Multinomial Naive Bayes: Resultados das métricas por algoritmo de vetorização

Vetorização	$\overline{Acc}$	$\overline{Prec}$	$\overline{Recall}$	$\overline{F1}$
TF-IDF	0.43	0.26	0.24	0.22
Bag of words	0.46	0.47	0.40	0.41
Bag of Words binária	0.52	0.56	0.45	0.46

Tabela 3 – Linear SVM: Resultados das métricas por algoritmo de vetorização

Vetorização	$\overline{Acc}$	$\overline{Prec}$	$\overline{Recall}$	$\overline{F1}$
TF-IDF	0.54	0.62	0.56	0.56
Bag of words	0.52	0.56	0.53	0.51
Bag of Words binária	0.58	0.65	0.60	0.59

apresentadas na Seção 2.4.4. Para fazer com que os resultados obtidos fossem menos dependentes da partição aleatória gerada, aplicamos o particionamento *holdout* 10 vezes e calculamos a média aritmética das medidas de desempenho delas. Desta forma, para cada partição aleatória de treino gerada, os algoritmos de vetorização textual irão gerar um novo vocabulário contendo os termos que estiveram presentes em exemplos da partição treino, e em seguida cada exemplo de treino será representado como um vetor numérico como descrito na seção 3.5.

Nas Tabelas 2 e 3 são exibidos os resultados obtidos na avaliação realizada utilizando os algoritmos Multinomial Naive Bayes e Máquina de vetores de suporte lineares respectivamente. Na primeira coluna de cada tabela são apresentados os métodos de vetorização utilizados; e nas colunas dois a cinco, são apresentadas as medidas de acurácia (Acc), precisão ($Prec$), revocação ($Recall$) e F1-score ($F1$). Podemos observar que a melhor acurácia e F1-score obtidos foram utilizando o algoritmo de aprendizado de máquina Máquinas de Vetores de Suporte Linear com o modelo de vetorização textual *bag of words* binária.

5 Conclusão

Neste trabalho, apresentamos uma metodologia para a categorização automática de conjuntos de dados de portais de dados governamentais abertos utilizando técnicas de processamento de linguagem natural e aprendizado de máquina supervisionado. Em nossa metodologia, utilizamos o conteúdo textual do nome do conjunto de dados e dos atributos dos arquivos anexados ao mesmo para inferir a categoria a qual ele pertence. Para a avaliação de nossa metodologia, utilizamos cinco portais das maiores cidades dos Estados Unidos que utilizam a plataforma CKAN como portal de dados abertos.

Os resultados obtidos em nossa avaliação da metodologia foram medidos usando medidas de desempenho de classificação em aprendizado de máquina, e indicam, entre outras métricas, uma acurácia de classificação de 58% quando utilizando como algoritmo de vetorização textual a *bag of words* binária e como algoritmo de aprendizado de máquina Máquinas de Vetores de Suporte Lineares. A quantidade ideal de conjuntos de dados por categoria, na fase de treinamento de um classificador, é uma questão em aberto que pode

contribuir para a obtenção de classificadores com maior acurácia.

5.1 Limitações

A extração de termos se mostra ineficiente em casos em que o atributo de um arquivo de dados possui um nome que consiste de duas ou mais palavras sendo unidas sem nenhum indicador claro, dependendo apenas da interpretação do leitor humano, por exemplo, “numerodefilhos” em que para nós é claro que significa “numero de filhos”.

A inferência de uma categoria usou como atributos de entrada apenas o nome do conjunto de dados e o nome das colunas dos seus arquivos anexados. O conteúdo das tuplas/registros dos arquivos de dados também poderiam ser utilizados como entrada para a indução de uma categoria.

5.2 Trabalhos futuros

Em nosso trabalho, por termos utilizado apenas conjunto de dados cuja categoria estava inclusa no conjunto de categorias que possuíam uma das palavras mais significativas em seu nome, muitos conjuntos de dados foram descartados. Um possível trabalho futuro seria, após a geração do classificador desenvolvido em nosso trabalho, inferir categorias do subconjunto abrangente para os conjuntos de dados descartados descritos anteriormente, e em seguida avaliar com a ajuda de voluntários se a classificação fez sentido ou não, sendo assim uma tarefa não-supervisionada de aprendizado de máquina.

Referências

BARBOSA, L. et al. Structured open urban data: understanding the landscape. *Big data, v. 2, n. 3*, p. 144–154, 2014. Citado na página 3.

FACELI, K. et al. *Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina*. [S.l.]: LTC, 2011. Citado 4 vezes nas páginas 5, 6, 7 e 11.

Frtunić Gligorijević, M. et al. Open data categorization based on formal concept analysis. *IEEE Transactions on Emerging Topics in Computing*, p. 1–1, 2019. ISSN 2376-4562. Citado na página 8.

INTERNATIONAL open knowledge. *The open definition*. <http://opendefinition.org/>. Acessado em: 2019-11-25. Citado na página 3.

INTERNATIONAL open knowledge. *open knowledge international*. <https://okfn.org/>. Acessado em: 2019-11-25. Citado na página 3.

INTERNATIONAL open knowledge. *What is Open?* <https://okfn.org/opendata/>. Acessado em: 2019-11-25. Citado na página 3.

LEARN scitkit. *scikit-learn*. 2019. <https://scikit-learn.org/>. Acessado em: 2019-11-25. Citado na página 8.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. *Introduction to Infromation Retrieval*. [S.l.]: Cambridge University Press, 2008. 258-262 p. Citado 3 vezes nas páginas 4, 5 e 6.

MANNING, C. D.; SCHüTZE, H. *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999. ISBN 0-262-13360-1. Citado na página 4.

M.F.PORTER. An algorithm for suffix stripping. *Program, 14(3)*, p. 130–137, 1980. Citado na página 5.

MIHALCEA, R.; CORLEY, C.; STRAPPARAVA, C. Corpus-based and knowledge-based measures of text semantic similarity. In: *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*. AAAI Press, 2006. (AAAI'06), p. 775–780. ISBN 978-1-57735-281-5. Disponível em: <http://dl.acm.org/citation.cfm?id=1597538.1597662>. Citado na página 8.

NLTK. *nltk*. 2019. <https://nltk.org>. Acessado em: 2019-11-25. Citado na página 5.

PINTO, H. dos S. Alinhamento de categorias em portais de dados abertos com base em um subconjunto abrangente. 2018. Citado 2 vezes nas páginas 8 e 11.

PINTO, H. dos S.; BERNARDINI, F.; VITERBO, J. How cities categorize datasets in their open data portals: an exploratory analysis. *dg.o 2018: Proceedings of the 19th Annual International Conference on Digital Government Research*, 2018. Citado 3 vezes nas páginas 2, 8 e 11.

PORTER, M. *The Porter Stemming Algorithm*. 2006. <https://tartarus.org/martin/PorterStemmer/>. Acessado em: 2019-11-26. Citado na página 5.

RAJARAMAN, A.; ULLMAN, J. *Data Mining: Mining of Massive Datasets*. [S.l.]: Cambridge University Press, 2011. 7–9 p. Citado na página 4.

YANG, H.-C.; LIN, C. S.; YU, P.-H. Toward automatic assessment of the categorization structure of open data portals. In: WANG, L. et al. (Ed.). *Multidisciplinary Social Networks Research*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. p. 372–380. Citado na página 8.

APÊNDICE A – Lista de tabelas

APÊNDICE B – Lista de ilustrações