CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

CLASSIFICAÇÃO DE NOTÍCIAS DE FRAUDE E CORRUPÇÃO EM PORTUGUÊS PARA INSTAURAÇÃO DE PROCESSO INVESTIGATIVO

Thiago Soares de Paula

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de mestre.

Orientadores:
Gustavo Paiva Guedes e Silva

Rio de Janeiro,

Outubro de 2022

Classificação de notícias de fraude e corrupção em Português para instauração de processo investigativo

Dissertação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Thiago Soares de Paula

Aprovada por:

Presidente, Prof. Gustavo Paiva Guedes e Silva, D.Sc. (orientador)

Eduardo Bezerra da Silva, D.Sc.

Ivandré Paraboni, D.Sc.

Rio de Janeiro,

Outubro de 2022

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

P324 Paula, Thiago Soares de
Classificação de notícias de fraude e corrupção em português para instauração de processo investigativo / Thiago Soares de Paula – 2022.
36f. : il. (algumas color.), tabs. ; enc.

Dissertação (Mestrado). Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2022.
Bibliografia : f. 34-36.
Orientador : Gustavo Paiva Guedes e Silva.

1. Mineração de dados (Computação). 2. Inteligência artificial. 3. Redes neurais (Computação). 4. Aprendizado do computador. 5. Mídia digital. I. Silva, Gustavo Paiva Guedes e (Orient.). II. Título.

CDD 006.312

Elaborada pelo bibliotecário Leandro Mota de Menezes CRB-7/5281

DEDICATÓRIA

Ao meu filho, Inácio, razão pela qual acordo todos os dias buscando melhorar.

À minha esposa, Clareana, por toda compreensão e parceria durante todo o período de ausência e dificuldades.

Ao meu pai, Reginaldo, e minha mãe, Martha, por terem me dado todo amor e suporte na formação acadêmica e pessoal.

À minha irmã, Priscila, por ter me inspirado a ser mais resiliente.

À minha sogra, Ana Cristina, por ter me ajudado nos momentos que precisei e ter cuidado do Inácio com tanto amor.

AGRADECIMENTOS

Agradeço principalmente a minha família por ter me apoiado nos momentos em que eu mais precisei.

Agradeço ao meu orientador, Gustavo Paiva Guedes e Silva, por toda a orientação, disponibilidade, paciência e parceria.

Agradeço a todo corpo docente do PPCIC, pelos ensinamentos.

Agradeço ao meu amigo Elbe Miranda, pela amizade e por toda ajuda técnica que precisei ao longo da pesquisa.

RESUMO

Classificação de notícias de fraude e corrupção em Português para instauração de processo investigativo

Thiago Soares de Paula

Orientador:
Gustavo Paiva Guedes e Silva

Resumo da Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do grau de mestre.

Os escândalos de fraude são fenômenos que podem gerar impactos imensuráveis nas esferas econômicas e reputacionais. Quando uma fraude é descoberta, os fatos normalmente vão a público por meio dos veículos de mídia, o que gera uma repercussão negativa muito grande. As empresas preocupadas com suas imagens têm investido cada vez mais esforços para minimizar ou atenuar os efeitos da fraude. Uma das tarefas que visa mitigar os efeitos da fraude é o monitoramento de mídias sobre fraude e corrupção. Essa tarefa é fundamental para a avaliação e o monitoramento dos riscos do negócio no mundo corporativo, pois a todo momento surgem fatos que podem trazer prejuízos à empresa e suas contrapartes. Uma vez veiculados escândalos de fraude em sites de notícias, os impactos podem gerar consequências negativas às imagens das empresas. Portanto, essas informações precisam ser coletadas e analisadas e, se necessário, encaminhadas para processo investigativo. No entanto, o grande volume de notícias publicadas por dia inviabiliza uma avaliação manual diária. Este trabalho apresenta uma abordagem que visa automatizar esse processo, o que inclui coletar notícias da web por meio de *web crawlers* dos principais veículos de mídias do Brasil, construir um corpus anotado em português sobre fraude e corrupção e criar um modelo de aprendizado de máquina cuja função é classificar notícias em relevantes ou não para abertura de investigação.

Palavras-chave:
Fraude, notícias, classificação

Rio de Janeiro,

Outubro de 2022

ABSTRACT

Classificação de notícias de fraude e corrupção em Português para instauração de processo investigativo

Thiago Soares de Paula

Advisors:
Gustavo Paiva Guedes e Silva

Abstract of dissertation submitted to Programa de Pós-graduação em Ciência da Computação - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

Fraud scandals are phenomena that can generate immeasurable impacts in the economic and reputational spheres. When a fraud is discovered, the facts are usually made public through the media, which generates a very large negative impact. Companies concerned about their images have invested more and more efforts to minimize or mitigate the effects of fraud. One of the tasks aimed at mitigating the effects of fraud is media monitoring of fraud and corruption. This task is fundamental for the assessment and monitoring of business risks in the corporate world, as facts arise that can cause harm to the company and its counterparties at all times. Once fraud scandals are reported on news sites, the impacts can have negative consequences for corporate images. Therefore, this information needs to be collected and analyzed and, if necessary, forwarded to the investigative process. However, the large volume of news published per day makes a daily manual assessment impossible. This work presents an approach that aims to automate this process, which includes collecting web news through web crawlers about the main media vehicles in Brazil, building an annotated corpus in Portuguese about fraud and corruption and creating a machine learning model whose function is to classify news as relevant or not for opening an investigation.

Key-words:
Fraud, News, Classification

Rio de Janeiro,

Outubro de 2022

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

AUC	Área Sobre A Curva	3
CA	Acurácia	3
CBOW	Continuos Bag Of Word	6
FN	Falso Negativo	10
FP	Falso Positivo	10
LR	Regressão Logística	2
NN	Rede Neural	2
RF	Random Forest	2
SVM	Support Vector Machine	2
VN	Verdadeiro Negativo	10
VP	Verdadeiro Positivo	10

Capítulo I Introdução

Os escândalos de fraude geram crises de credibilidade e prejuízos financeiros imensuráveis para as empresas envolvidas nesses eventos. Os efeitos negativos atingem não somente as empresas fraudadoras como também as que sofrem a fraude. Os impactos são tão grandes que afetam também a clientes, acionistas e toda uma cadeia econômica ligada as companhias fraudadoras [Quirk, 1997]. Podemos citar algumas gigantes mundiais como Enron, Volkswagen e Glogal Crossing que sofreram grandes impactos de escândalos relacionados a fraude, assim como Petrobras, Odebrech, JBS, que no brasil passaram por crises enormes de credibilidade e até hoje lutam para recuperar suas imagens [Costa and Wood Jr, 2012].

Com o mundo corporativo cada vez mais competitivo, as empresas vêm se voltando para implantação de iniciativas preventivas e detectivas que visam minimizar ou até mesmo evitar os efeitos da fraude [Bolton and Hand, 2002]. Várias medidas vêm sendo adotadas e aperfeiçadas desde soluções mais simples, como marcas d'água que dificultam a falsificação de códigos de barra de boletos, até modelos sofisticados baseados em aprendizagem de máquina para detecção de transações financeiras suspeitas. A evolução desses métodos se faz necessária, pois os fraudadores, ao conhecerem as técnicas de detecção, criam novos artifícios para burlar esses controles [Bolton and Hand, 2002].

I.1 Motivação

Na linha detectiva, existem algumas medidas que podem ser adotadas para minimizar os impactos causados por esses escândalos e uma delas é a pesquisa de mídias sobre fraude e corrupção na web. Essa atividade pode ser utilizada para monitorar fatos relevantes relacionados à empresa, pois é sabido que os eventos de fraude, assim que chegam ao conhecimento dos veículos de mídia, tomam proporções gigantescas, gerando um volume de informações enorme nos sites de notícias. Toda essa informação gerada, se utilizada pelas empresas de forma adequada, pode minimizar os impactos negativos, pois permite que sejam tomadas medidas que podem gerar uma boa resposta pelo mercado e consequente impacto positivo na imagem da companhia [Gottschalk, 2016]. Todo esse trabalho de pesquisa visa identificar fatos relevantes relacionados a fraude que estão associados com os negócios da empresa e subsidiar análises sobre o impacto positivo ou negativo da informação,

permitindo o mapeamento de riscos e até mesmo a abertura de uma investigação interna. Para isso, as equipes responsáveis pelo monitoramento de mídias de fraude e corrupção precisam coletar, analisar e classificar essas notícias em relevantes ou não para investigação. Todo esse processo de coleta e classificação de notícias é efetuado manualmente e gera uma demanda muito grande por recursos, tendo em vista que o volume de informações publicadas diariamente é muito grande, o que torna as atividades de leitura e classificação dessas informações inviáveis de serem realizadas manualmente e com equipe reduzida [de Brito, 2018].

Com objetivo de aumentar a eficiência do trabalho de monitoramento, melhorar a cobertura das notícias e mitigar os riscos inerentes aos negócios, como eventos de fraude e corrupção entres seus colaboradores e parceiros de negócio, as empresas têm investido cada vez mais no monitoramento automático de mídias sobre fraude e corrupção para identificar fatos relevantes [Rasekh, 2015]. Tendo em vista essas necessidades, alguns trabalhos foram desenvolvidos [Thaipisutikul et al., 2021; Weichselbraun et al., 2020; Liu et al., 2020] com objetivo de coletar e classificar automaticamente as notícias veiculadas na web, por meio de modelos de classificação baseados em aprendizado de máquina. Entretanto, não foram encontradas abordagens cujos modelos utilizaram corpus em Português.

I.2 Objetivo

Sendo assim, esta pesquisa visa responder a seguinte pergunta de pesquisa: é possível automatizar o processo de monitoramento e classificação de notícias sobre fraude e corrupção para sugerir a instauração de processo investigativo? A partir desse questionamento, este trabalho apresenta uma proposta com objetivo de automatizar o processo de monitoramento de notícias sobre fraude e corrupção em Português, o que inclui coletá-las e classificá-las em relevantes ou não para abertura de processo investigativo interno. Para isso, inicialmente foram coletadas aproximadamente 1 milhão de notícias (da web) dos principais veículos de mídias do Brasil, utilizando *web clawlers* desenvolvidos com o *framework scrapy*. Essas notícias serviram de insumo para a construção de um corpus anotado (em português) de notícias sobre fraude e corrupção (FraudeCorpusBR). Em seguida, foram treinados 4 classificadores - Rede Neural (NN), *Random Forest (RF)*, *Support Vector Machine (SVM)* e Regressão Logística (LR) - com objetivo de comparar o desempenho de diferentes modelos na tarefa de classificar notícias de fraude e corrupção para abertura de processo investigativo. Por fim, visando melhorar os resultados do modelo, foi treinado um modelo *ensemble* que utiliza como entrada a saída dos 4 modelos anteriores.

I.3 Experimentos realizados

Nesta etapa foram realizados experimentos para comparar o desempenho dos classificadores, que utilizou o FraudeCorpusBR, um conjunto de *features* gerado com base nesse corpus e quatro classificadores - Rede Neural (NN), *Random Forest*(RF), *Support Vector Machine*(SVM) e Regressão Logistica(RL). As features geradas foram submetidas a treinamento dos quatro classificadores. Em seguida foram efetuadas algumas avaliações considerando as métricas Acurácia (CA), Área sobre a curva (AUC), Precisão, *F1 Score* e *Recall* e foi possível observar o comportamento dos modelos utilizados com base nos conjuntos de *features* individualmente e combinados.

I.4 Organização

Além do capítulo I que discorre sobre a introdução da dissertação, o restante do trabalho está organizado em mais 6 capítulos que falam do referencial teórico, trabalhos relacionados, conjunto de dados, metodologia, experimentos e conclusões. O capítulo II é abordado todo o referencial teórico utilizado para suportar este trabalho como conceitos, motivações, ferramentas e contribuições relacionadas ao tema de classificação de notícias.

O capítulo III traz informações sobre os trabalhos desenvolvidos relacionados ao tema classificação de notícias sobre fraude e corrupção. Em função do número de trabalhos reduzidos na literatura sobre o tema dessa dissertação, serão apresentados também nesse capítulo os trabalhos sobre classificação de notícias de temas gerais. São descritos os idiomas utilizados nos trabalhos, os classificadores, o desempenho e as contribuições de cada um.

O capítulo IV apresenta a primeira das duas contribuições principais desse trabalho que é o conjunto de dados FraudeCorpusBR. Será apresentado também como se deu a construção do corpus anotado de forma automática (FraudeCorpusBR-A) que tornou o trabalho de anotação mais eficiente.

O capítulo V apresenta a segunda e mais importante contribuição desse trabalho que é o modelo de classificação de notícias em português sobre fraude e corrupção para abertura de processo investigativo. Foi descrito nesse capítulo os tipos de classificadores utilizados, as *features* criadas e como os resultados foram melhorados utilizando o *ensemble* de classificadores.

O capítulo VI apresenta os experimentos que foram desenvolvidos. Neles utilizamos combinações de corpus, *features* e classificadores e ao final são apresentadas tabelas que permitem comparar quais modelos tiveram os melhores resultados para cada combinação (corpus, *features* e classificadores).

O capítulo VII traz as conclusões sobre o trabalho, em que são apresentados os resultados obtidos até o momento, além de mostrar algumas dificuldades encontradas ao longo da execução dos experimentos. Por fim, são descritos também aspectos sobre o futuro do trabalho.

Capítulo II Referencial teórico

Neste capítulo é apresentado o referencial teórico para a compreensão da metodologia proposta nesta dissertação. Serão apresentados aqui conceitos, motivações, modelos e contribuições relacionadas ao tema de Classificação de notícias sobre fraude e corrupção. Serão abordados o processo de monitoramento de mídia, o modelo de aprendizagem de máquina *word2vec*, o método de similaridade do cosseno e os classificadores *Support vector machine*(SVM), *Random forest*, Rede Neural, Regressão Logística, *Naive bayes* (NB), *K-Nearest Neighbors* (KNN), Árvore de decisão e *AdaBoost*

II.1 Monitoramento de Mídias

O monitoramento de mídias externas tem a função de checar os sites de notícias com o objetivo de detectar algum fato relevante sobre a empresa e suas contrapartes. Caso seja identificado algum fato relevante que venha colocar em risco a reputação da empresa, o processo de monitoramento de mídia se encarrega de encaminhar os fatos para investigação. Todo esse processo pode ser visto na figura II.1. Essa é uma importante ferramenta de Inteligência Competitiva, uma vez que permite que as empresas se antecipem a fatos que podem impactar seus negócios [Tan et al., 2002].

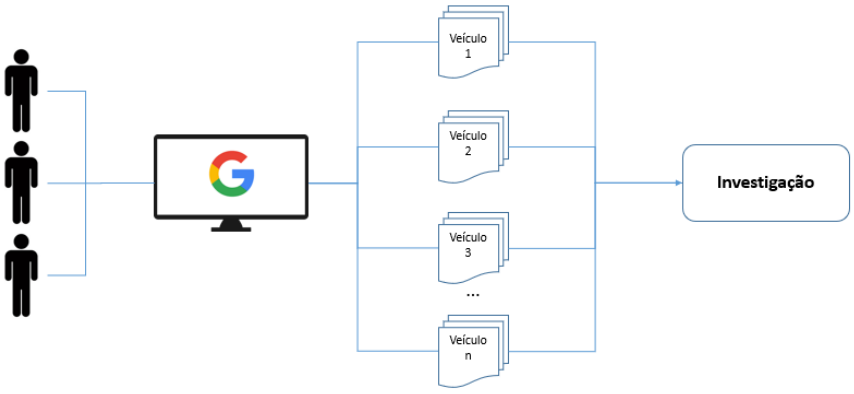

Figura II.1: Monitoramento de Mídias

II.2 Web crawler e Web Scraping

Quando nos referimos a extração automática de dados da web, dois conceitos surgem: *Web Scraping* e *Web Crawling*. A tarefa de *Web Crawling* é realizada pelas ferramentas de busca para indexar as páginas da web, em que um "robô" vai visitando as páginas web e seguindo cada um

dos *links* presentes nelas, como se estivéssemos "rastejando pela teia" [Amudha and Phil, 2017]. No caso do *Web Scraping*, estamos mais interessados em extrair os dados de uma página em particular, por isso, precisamos estudar a estrutura HTML da página e identificar as *tags* cujos dados queremos extrair, como se estivéssemos "raspando" os dados da página [Sundaramoorthy et al., 2017]. Normalmente essas técnicas são utilizadas em conjunto. O *Web Scraping* é usado para extrair os dados de uma notícia, como título, data e texto, e o *Web Crawling* para percorrer a paginação das notícias e extrair os links delas. Um dos frameworks mais famosos utilizados nos processos de *Web Scraping* e *Web Crawling* é o *Scrapy* e sua arquitetura pode ser vista na figura II.2

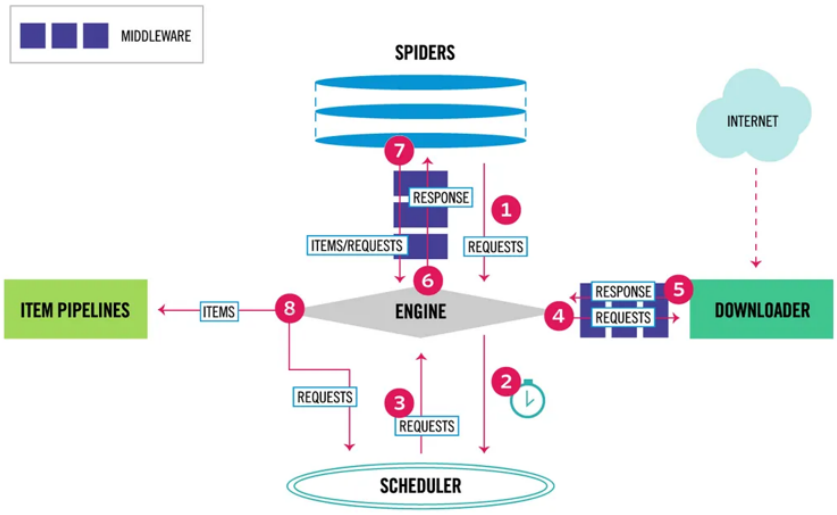

Figura II.2: Arquitetura Scrapy [Sundaramoorthy et al., 2017]

II.3 Word2Vec

O *Word2Vec* é um modelo que visa capturar a relação semântica de cada palavra. Ele implementa uma abordagem chamada *word embeddings*, que representa palavras por meio de vetores numéricos e permite identificar o relacionamento semântico a partir das propriedades observadas no corpus de treinamento. É possível também extrair a relação entre as palavras, como por exemplo: "londres" está para "Inglaterra" assim como "Paris" está para "França". É possível observar na figura II.3 como as palavras são representadas a partir dos seus contextos.

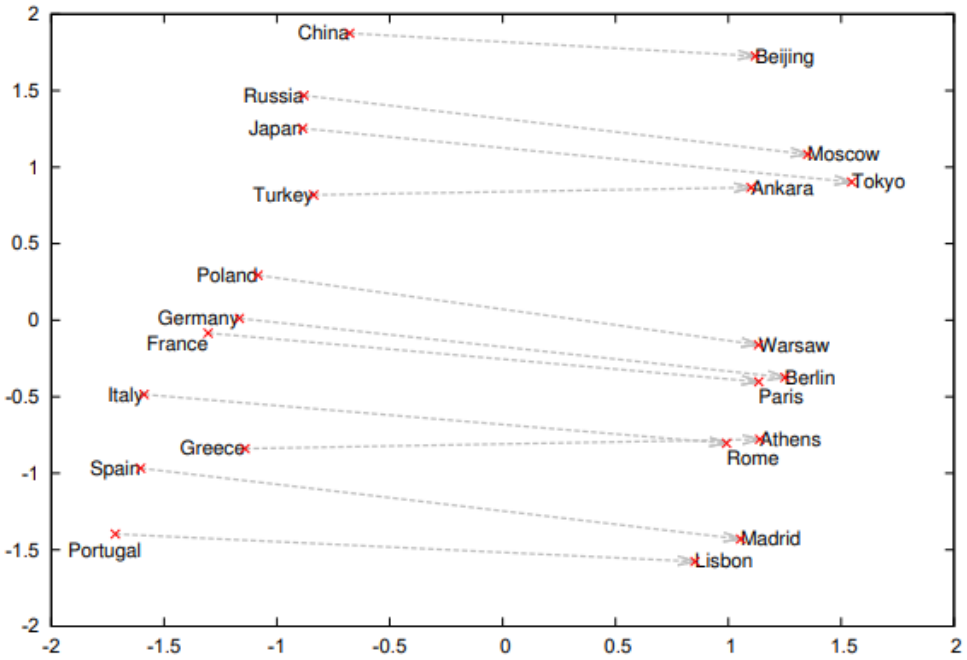

Figura II.3: *Word2vec* [Mikolov et al., 2013]

O *Word2Vec* é treinado com uma rede neural e pode utilizar duas abordagens conforme representado na figura II.4. São elas: *Continuos Bag of Word (CBOW)* e *Skip-gram*. A ideia do *CBOW* é prever qual palavra estamos buscando a partir de um determinado contexto. Em contrapartida, o *Skip-gram* retorna o contexto a partir de uma palavra [Mikolov et al., 2013].

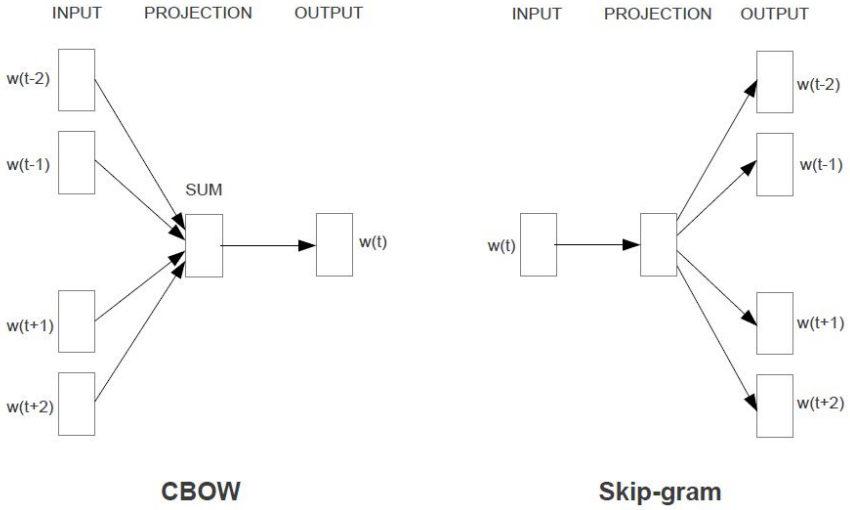

Figura II.4: Skip-gram e CBOW [Mikolov et al., 2013]

II.4 Similaridade do cosseno

A similaridade do cosseno consiste em calcular a similaridade entre dois vetores num espaço vetorial, que avalia o valor do cosseno do ângulo compreendido entre eles. Esta função produz resultado 1 se o ângulo compreendido é zero, ou seja, se os vetores apontam pra mesma direção

[Elberrichi et al., 2008]. A similaridade do cosseno é calculada pelo produto escalar de dois vetores divido pelo produto entre os módulos conforme equação II.1.

$$Similarity = \cos(\theta) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \tag{II.1}$$

II.5 Engenharia de feature

O objetivo da engenharia de *feature* é selecionar o melhor conjunto de *features* para extrair o melhor resultado do modelo de aprendizagem de máquina. Todos os modelos de aprendizagem de máquina devem receber entradas, processá-las e produzir uma saída, seja na tarefa de classificação ou regressão. As entradas dos modelos se chamam *features*. Muitas vezes é necessário gerar novas *features* e até mesmo fazer a transformação matemática das existentes, pois os *datasets* não possuem *features* suficientes e nos formatos adequados para um resultado satisfatório do modelo. O processo de Engenharia de *feature* envolve a análise, transformação e geração de novos dados para extrair o melhor desempenho do modelo.

II.6 Classificadores

II.6.1 Support vector machines (SVMs)

Support vector machines (SVMs) são um conjunto de métodos de aprendizado de máquina supervisionado usados para criação de modelos para as tarefas de classificação, regressão e detecção de *outlier* [Mammone et al., 2009]. O SVM recebe um conjunto de dados como entrada e, para cada entrada, prediz qual das duas possíveis classes ela faz parte. O SVM é um classificador linear binário e funciona definindo uma linha de separação (hiperplano) entre os dados das classes possíveis. Esse limite busca maximizar a distância entre os pontos mais próximos em relação a cada uma das classes. A distância entre o hiperplano e primeiro ponto de cada classe é denominado de margem. É um algoritmo muito adequado para problemas de classificação de texto, em que é comum ter acesso a um conjunto de dados rotulado reduzido [Mammone et al., 2009].

II.6.2 Random Forest

Random Forest é um método de aprendizagem supervisionada que pode ser utilizado tanto para classificação quanto para regressão. Ele se baseia em ensemble do tipo *bagging*, em que várias árvores de decisões são construídas em tempo de treinamento. Para a tarefa de classificação, o resultado é calculado pela classe que recebe a maior quantidade de votos pelas árvores. No caso de regressão, a saída é gerada a partir da média das saídas das árvores [Livingston, 2005].

II.6.3 Regressão logística

Regressão logística é uma técnica que tem como objetivo efetuar a predição de dados categóricos (normalmente binários), a partir da observação de um conjunto de variáveis categóricas e contínuas. É possível estimar a probabilidade da ocorrência de um determinado evento em face de variáveis explanatórias. É uma técnica análoga a regressão linear, mas é utilizada para problemas de classificação [Palei and Das, 2009].

II.6.4 Naive Bayes

Naive bayes é um classificador probabilístico muito utilizado em aprendizado de máquina para categorizar textos baseado na frequência das palavras. O modelo foi criado por um matemático inglês e se baseia no *Teorema de bayes*. O algoritmo recebe o nome de *Bayes*(ingênuo) porque desconsidera a relação entre as variáveis e cada *feature* é tratada de maneira totalmente independente [Lewis, 1998]. Por ser matematicamente simples, possui bom desempenho e precisa de poucos exemplos para ter uma boa acurácia. O algoritmo é muito utilizado para prever diagnósticos de doenças, análise de crédito, determinar o risco de um email ser *spam* dentre outras aplicações [Lewis, 1998].

II.6.5 K-Nearest Neighbors (KNN)

O algoritmo KNN tem como objetivo identificar os K vizinhos mais próximos de uma documento especifico entre todos os documentos classificados. Cada documento é representado por um vetor com N características em um espaço dimensional. Esses vetores são utilizados para calcular a distância euclidiana entre documentos que é utilizada para determinar a similaridade entre documentos.

II.6.6 Árvore de decisão

É um algoritmo aplicado em tarefas de classificação e regressão e utiliza a estratégia de "dividir para conquistar", em que um problema complexo é divido em problemas mais simples até que a tarefa maior seja resolvida. O objetivo do algoritmo é criar uma árvore que aprende com os dados por meio de regras básicas. É identificado qual o melhor atributo para separar os dados utilizando o critério entropia ou gini. O critério de gini isola um ramo com registros que representam a classe mais frequente. Enquanto a entropia balanceia os registros em cada ramo.

II.6.7 AdaBoost

É um método que utiliza a combinação de diversos modelos de aprendizagem fracos para formar um modelo de aprendizagem mais forte. O objetivo principal é treinar preditores sequencialmente com objetivo de cada um corrigir seu antecessor.

II.6.8 Rede neural

Rede Neural são sistemas computacionais com nós interconectados que funcionam semelhantes aos neurônios humanos. Algoritmos baseados em rede neural são capazes de identificar padrões escondidos e correlações em dados brutos, agrupá-los e classificá-los [Simons, 2009]. A rede neural tem a capacidade de aprender e melhorar seu desempenho com base nas predições produzidas. As redes neurais artificiais são compostas por camadas, contendo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída [Simons, 2009]. Cada nó, ou neurônio artificial, conecta-se a outro e tem um peso e um limite associados [Simons, 2009]. Se a saída de qualquer nó individual estiver acima do valor do limite especificado, esse nó será ativado, enviando dados para a próxima camada da rede [Simons, 2009]. Caso contrário, nenhum dado será transmitido para a próxima camada da rede.

II.6.9 Ensemble learning

Ensemble learning ou *Aprendizado por Comitê* é um tipo de método de aprendizagem que utiliza vários algoritmos de forma que combinados, produzam um resultado melhor se comparados aos resultados individuais de cada modelo [Opitz and Maclin, 1999]. Existem dois aspectos importantes quando estamos treinando um modelo de aprendizagem de máquina que são o viés e a variância. O *ensemble* ajuda a encontrar o equilíbrio entre essas duas variáveis [Opitz and Maclin, 1999]. As três formas principais de *ensemble* são:

1. *Bagging*: Utiliza diferentes modelos base, treinados em paralelo com amostras reduzidas do *dataset* (similares entre si) e gera um resultado utilizando a média das saídas de cada modelo com objetivo de reduzir a variância. Para modelos baseados em regressão, gera a média das predições. Já no caso dos modelos de classificação, a saída é gerada com base na classe mais votada ou na média das probabilidades dos resultados da classificação [Opitz and Maclin, 1999].

2. *Boosting*: Os modelos são treinados de maneira sequencial, de forma que os erros da saída do modelo anterior são utilizados no próximo modelo a ser treinado e os pesos matemáticos sejam sempre ajustados com base no modelo anterior [Opitz and Maclin, 1999]. Diferentemente do *Bagging*, o *Boosting* foca na redução do viés.

3. *Stacking*: Funciona construindo modelo mais robustos utilizando as predições de modelos mais fracos como *features*. Essas novas features permitem melhorar os modelos finais nos aspectos que os modelos inicias (fracos) tiveram desempenhos ruins [Opitz and Maclin, 1999].

II.7 Métricas de avaliação

São indicadores que ajudam a avaliar a qualidade de modelo de aprendizado de máquina. Existem métricas mais simples e outras mais complexas, cada uma adequada para um tipo de avaliação. A escolha de uma métrica de avaliação passa por fatores como proporção dos dados do conjunto de dados e objetivo da predição. No que se refere aos modelos de classificação, as predições realizadas pelo modelo podem ser:

1. Verdadeiro Positivo (VP): Exemplo que o modelo previu ser verdadeiro e realmente era verdadeiro
2. Verdadeiro Negativo (VN): Exemplo que o modelo previu ser verdadeiro e não era verdadeiro
3. Falso Positivo (FP): Exemplo que o modelo previu ser falso e realmente era falso
4. Falso Negativo (FN): Exemplo que o modelo previu ser falso e não era falso

II.7.1 Acurácia (CA)

A acurácia é uma métrica que mostra, dentre todas as classificações, quantas o modelo classificou corretamente. É a métrica mais simples e mostra o percentual de acertos do modelo. É gerada a partir do cálculo da razão entre a quantidade de acertos e total de entradas, conforme a equação II.2. Ou seja, quantos exemplos o modelo previu ser de uma classe (positivo e negativo) que realmente eram daquela [Han et al., 2022].

$$Acurácia = \frac{VP + VN}{VP + FN + VN + FP} \tag{II.2}$$

II.7.2 Precisão

Apresenta um índice que mostra dentre todos os valores positivos quantos efetivamente o modelo acertou em dizer que eram efetivamente verdadeiros. A Precisão refere-se a quantidade de verdadeiros positivos (VP) sobre a quantidade total de predições positivas. Normalmente é usada quando os falsos positivos são mais prejudiciais que os falsos negativos [Han et al., 2022].

$$Precisão = \frac{VP}{VP + FP} \tag{II.3}$$

II.7.3 Recall

A métrica de *recall* dá uma enfase maior para erros falsos negativos. É definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e quantidade de exemplos que são de fato positivos, conforme equação II.4 [Han et al., 2022].

$$Recall = \frac{VP}{VP + FN} \tag{II.4}$$

II.7.4 Área sobre a curva (AUC)

O AUC indica a medida de separabilidade das classificações efetuadas pelo modelo. Quanto maior o AUC, melhor o modelo está em prever falso como falso e verdadeiro como verdadeiro. Por exemplo, quanto maior a AUC, melhor o modelo está em distinguir entre emails que são spam dos que não são. O valor do AUC varia de 0.0 até 1.0 e permite demonstrar o desempenho de um modelo de aprendizagem de máquina, que seja um classificador binário, por meio da relação da taxa de verdadeiro positivo e da taxa de falso positivo, variando o limiar [Han et al., 2022].

II.7.5 F1 score

A métrica *F1 score* ou *F1-measure* é um indicador que leva em consideração as medidas de Precisão e Recall. É uma medida que é definida pela média harmônica entre Precisão e Recall. Um fator importante desse indicador é que se o recall e a precisão forem baixos o F1 será baixo. Assim se ambos forem altos, o F1 também será. Outra característica importante é que se só uma das métricas for alta, o F1 não será alto. Isso auxilia na escolha de um modelo de classificação que seja mais equilibrado [Han et al., 2022]. A fórmula do F1 score é apresentada na equação II.5.

$$F1score = 2 * \frac{Precisao * Recall}{Precisao * Recall} \tag{II.5}$$

II.8 Abordagens para solucionar problemas de classes desbalanceadas

Essas abordagens tem o objetivo de minimizar as discrepâncias entre as classes do *dataset* que geram desbalanceamento. São aplicadas técnicas que realizam a reamostragem das classes com objetivo de balancear o *dataset* [Han et al., 2022]. O desbalanceamento pode ser resolvido utilizando as técnicas de descarte (*Under-sampling*) e duplicação (*Over-sampling*) de exemplos. A técnica de Under-sampling elimina aleatoriamente exemplos da classe majoritária para reduzir o desbalanceamento do *dataset* [Dal Pozzolo et al., 2015] [Han et al., 2022]. Já a técnica de Over-sampling cria novos exemplos da classe minoritária a partir das características presentes nos dados

originais. A geração de novas entradas pode ser feita aleatoriamente ou sinteticamente [Han et al., 2022].

Capítulo III Trabalhos Relacionados

Este capítulo descreve os trabalhos relacionados ao tema de classificação de notícias sobre fraude e corrupção. Como não existem até o momento muitos trabalhos publicados na literatura sobre esse tema, foram relacionados também trabalhos sobre o tema classificação de notícias sem o foco em fraude e corrupção.

III.1 Trabalhos sobre classficação de notícias de fraude e corrupção

Thaipisutikul et al. [2021] desenvolveram um sistema com objetivo de detectar eventos por meio do monitoramento automático de notícias Tailandesas. As notícias são classificadas em categorias como roubo, droga, assassinato, acidente e corrupção. Além da classificação efetuada, são identificadas as localizações e o tempo em que as notícias ocorreram. O sistema proposto pode ser utilizado por órgãos responsáveis pela aplicação das leis, uma vez que permite uma atuação preventiva e estratégica para mitigar atos criminais no futuro. Além disso, o monitoramento automático de notícias pode ajudar as pessoas a tomarem conhecimento sobre crimes que acontecem em suas regiões. O trabalho de monitoramento é executado por meio da coleta de notícias em fontes públicas utilizando *web crawlers*. Essas notícias são pré-processadas em vetores gerados a partir da aplicação da abordagem de *Term Frequency-Inverse Document Frequency* (TF-IDF). Por fim, esses vetores são submetidos para treinamento de 6 classificadores. São eles: *Multinomial Naive Bayes*, *Gradient Boosting Machine*, *Random Forest*, *K Nearest Neighbors*, *Multinomial Logistic Regression* e *Support Vector Machine* (SVM). O objetivo principal do trabalho foi mostrar um comparativo dos classificadores, apresentando quais produziram as melhores métricas de acurácia, *recall* e *F-measure*. Por fim, o algoritmo que apresentou o melhor resultado foi o SVM com 0.81 de precisão, 0.78 de *recall* e 0.80 de *F-measure*.

Weichselbraun et al. [2020] apresentaram um *dashboard* chamado de Monitor de Risco de Integridade, criado com base em técnicas de *Web Intelligence* e *Deep Learning*, que se propõe a classificar textos em inglês e alemão para determinar o risco de integridade para o negócio. A proposta usa documentos anotados para rastrear e visualizar lacunas de gerenciamento de integridade anteriores e seus respectivos impactos, identificar se as organizações foram mencionados positivamente ou negativamente nas notícias, utilizar a base histórica de notícias para prever novos eventos

negativos e detectar pontos cegos existentes dentro de uma empresa. Por meio dos termos presentes nos documentos analisados, também é possível identificar geograficamente quais regiões apresentam maior incidência de eventos relacionados a riscos de integridade. Para desenvolver esse painel, foram utilizados dados históricos para treinamento dos classificadores *Support Vector Machines*, *Naive Bayes*, *Long Short Term Classifier*, *Bidirectional Long Short Term Classifier* e *Convolutional Neural Network*. Também foram utilizadas as *features bag of words* e dois tipos de *embeddings*. O classificador que atingiu o melhor resultado (F1 de 0.8778) foi uma rede convolucional utilizando como *feature* os *embeddings* dos textos.

Lima et al. [2020] apresentaram uma proposta para prever os riscos nos processos de contratação do governo brasileiro por meio da coleta dos textos dos editais de licitações disponíveis em meios públicos. Foi produzido nesse trabalho um novo conjunto de dados formado por editais de licitações disponíveis no Diário Oficial da União, utilizando 15.132.968 entradas textuais, das quais 1.907 são entradas de risco anotadas. Durante os experimentos efetuados, tanto a *bottleneck deep neural network* quanto o *Bi-LSTM* mostraram-se competitivos em relação aos classificadores clássicos e obtiveram melhor precisão (93,0% e 92,4%, respectivamente).

Liu et al. [2020] apresentaram um modelo de classificação de texto para o mercado de capital chinês cujo objetivo é identificar empresas fraudulentas. Foram extraídas *features* como TF-IDF, tópicos e quantidades de notícias de escândalos relacionados à empresa. Essas *features* foram criadas a partir de conteúdos gerados por usuários de redes sociais. O trabalho teve como base a teoria da linguística funcional sistêmica e tem como objetivo contribuir para determinação de riscos em transações realizadas por investidores individuais, institucionais e órgãos reguladores.

III.2 Trabalhos sobre classificação de notícias

Kumar et al. [2021] apresentaram um modelo de aprendizado de máquina baseado em *Multinomial Naive Bayes* (MNB) com a *feature Inverse Document Frequency* (IDF) para classificar notícias indianas de assuntos atuais em categorias e que atingiu precisão de 87.22 por cento. As categorias consideradas pelo modelo foram Negócios e Economia, Educação e carreira, entretenimento, Alimentação e saúde, internacional, Política e Governança, Ciência e Tecnologia e esportes. O modelo MNB com as *features Count Vectorized* e IDF também apresentaram os melhores resultados em relação a eficiência com 1.82s e 3.86s, respectivamente. Além disso, é apresentado um comparativo dos resultados do MNB com classificadores como Regressão Logística (LR), *Vetor de Suporte Máquina* (SVM), *K Nearest Neighbor* (KNN) e *Random Forest* (RF). Os resultados demonstram que o modelo baseado em MNB é melhor em termos de precisão e pode ser implantado na tarefa de classificação de informações visando melhorar a produtividade e a eficiência na classificação de notícias.

Junardi and Khodra [2020] apresentaram um modelo para classificação de notícias sobre a indústria para auxiliar nas explicações e análises econômicas no PIB (Produto Interno Bruto) da Indonésia. O modelo se propõem substituir o processo manual de análise de notícias e classificação por categoria que é utilizado na elaboração do PIB Indonésio. Foi usado um método de classificação multi-rótulo, pois cada item de notícia tem uma ou mais categorias. O rótulo corresponde ao PIB por setor, que se refere ao padrão indonésio de classificação industrial (KBLI). Foram usados os classificadores *Linear SVC*, *Random Forest* (RF), *Naive Bayes* (NB), Regressão logística (LR), Árvore de decisão (DT), e *K-Nearest Neighbors* (KNN). O método *Label Power-set* combinado com *Linear SVC* mostrou o melhor resultado dentre todos e alcançou 75% para *F-Measure*.

Calomiris and Mamaysky [2019] desenvolveram uma metodologia de classificação de artigos de notícias com o objetivo de determinar o risco e o retorno em mercados de ações em 51 economias de países emergentes considerando dados do período de 1998 a 2015. O método proposto considera as características de cada país e os impactos que as notícias tem sobre os resultados das ações no mercado. Para isso, são considerados aspectos como o sentimento da palavra (negativo ou positivo), frequência em que ocorrem e o contexto em que se encontram. Foi possível determinar que o fluxo de palavras captura informações implícitas das notícias que não são compreendidas no momento em que os artigos aparecem, mas que captam influências no mercado que têm relevância crescente ao longo do tempo.

Hallac et al. [2018] apresentaram uma abordagem para classificação de texto de *tweets* para identificar se um *tweet* pertence aos tópicos de cultura, economia, política, esportes ou tecnologia. A abordagem mostra que é possível classificar os *tweets* em níveis de alta precisão mesmo com uma quantidade muito pequena de dados rotulados. Para isso, uma rede neural simples foi treinada com quantidade grande de textos de notícias. Em seguida, foram aplicadas etapas básicas de ajuste fino no modelo usando dados de *tweet*. Foram utilizados os classificadores Rede Neural Convolucional (CNN), Redes Neurais Recorrentes (RNN) e *Bi-LSTM-CONV*, que receberam como *feature* os *embeddings* extraídos dos textos. O classificador que obteve o melhor foi o *Bi-LSTM-CONV* com 83.20% de acurácia.

Gürcan [2018] apresentou uma proposta para classificar os textos turcos com base em modelos de aprendizado de máquina supervisionado. Esses modelos foram treinados para efetuar a classificação de textos de notícias em cinco classes predefinidas (economia, política, esporte, saúde e tecnologia) e utilizou uma variedade de documentos para o processo de classificação. O trabalho também mostra o comparativo dos desempenhos dos modelos de classificação *Multinomial Naive Bayes*, *Bernoulli Naive Bayes*, Suporte Máquina de vetores (SVM), *K Nearest Neighbor* (KNN) e o algoritmo de árvore de decisão em textos de notícias turcos. Esses desempenhos também são interpretados à luz dos resultados obtidos com diferentes parâmetros. Por fim, o modelo com o melhor resultado na

tarefa de classificação foi o *Multinomial Naive Bayes* com a acurácia de 90%.

Li-Juan et al. [2015] propõe um método que seleciona as entidades nomeadas e palavras-chave em títulos de notícias vietnamitas e adota o modelo de entropia máxima para alcançar classificação. Além disso, uma das contribuições principais do trabalho é a coleta mais de 6.000 textos de notícias vietnamitas, que são rotuladas em sete tipos de categorias de notícias como política, economia e cultura, etc. O trabalho realizou experimentos que mostram que a precisão do método de classificação de notícias vietnamitas alcançou 96,97%.

Cabe destacar que embora praticamente todos os trabalhos relacionados tenham apresentado abordagens para coletar e classificar notícias, nenhum deles trabalhou com corpus em Português. Outro aspecto importante é que não foram identificados trabalhos na literatura relacionados a tarefa proposta nessa dissertação. Destacamos também que o modelo atingiu uma *F-measure* de 93,4%.

Capítulo IV Conjunto de dados

Esta seção descreve todo o processo de construção do corpus de notícias em português sobre fraude e corrupção (FraudeCorpusBR), que consistiu na coleta, filtragem e anotação das notícias. Também será abordada a descrição das etapas de anotação manual, efetuada por um grupo de especialistas e a anotação automática, efetuada agrupando notícias por similaridade utilizando o modelo de linguagem word2vec pre-treinado em português.

IV.1 FraudeCorpusBR

A construção do FraudCorpusBR seguiu 7 etapas, conforme ilustrado na figura IV.1. Na primeira etapa foram extraídas aproximadamente 1 milhão de notícias (da web) de 45 veículos brasileiros de notícias relacionadas à economia e política utilizando *web clawlers* desenvolvidos com o *framework scrapy*. A segunda etapa foi responsável pela filtragem do conjunto inicial de notícias, mantendo-se somente aquelas notícias que continham pelo menos um dos termos de uma lista de 44 termos (FraudTerms)[1] de fraude e corrupção, resultando em um total de 335 mil notícias. Para isso, foi utilizada uma expressão regular que contou com as variações dos termos contidos na lista *FraudTerms* e não somente os termos exatos. A lista FraudTerms foi definida com base nos termos mais utilizados nas pesquisas manuais efetuadas por equipes de uma empresa estatal do governo brasileiro responsável pelo monitoramento de mídias sobre fraude e corrupção.

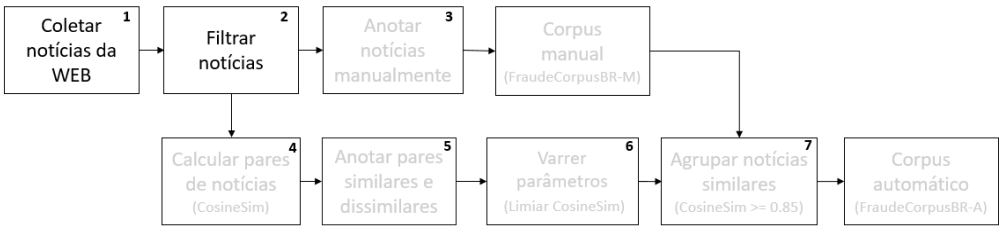

Figura IV.1: Etapas da construção do corpus de notícias em português

[1]Termos de fraude: corrupção, fraude, propina, favorecimento, receber vantagem, oferecer vantagem, mal feito, lavagem de dinheiro, nepotismo, conflito de interesse, caixa-dois, ilícito, vantagem indevida, suborno, dinheiro sujo, roubo, desfalque, desonesto, abuso, impunidade, má administração, má gestão, crime, prevaricação, má conduta, extorsão, criminoso, uso indevido, evasão de divisa, evasão fiscal, tráfico de influência, tráfico de droga, transgressão, violação, conspiração, ilegalidade, máfia, facção, apropriação indébita, chantagem, furto, cleptocracia, tráfico de drogas, milícia.

IV.1.1 FraudeCorpusBR-M

A partir do conjunto das 335 mil notícias filtradas, iniciou-se a etapa 3, responsável pelo processo de anotação manual. Esse processo foi realizado por 6 juízes diferentes, especialistas na atividade de monitoramento de mídias sobre fraude e corrupção. Cada juiz especialista anotou aproximadamente 376 notícias e não houve sobreposição de notícias anotadas, ou seja, cada notícia foi anotada por um especialista. Para auxiliar o processo de anotação, foi desenvolvido um *web form*, conforme ilustrado na figura IV.2, em que uma notícia era selecionada aleatoriamente do conjunto de 335 mil para anotação do juiz.

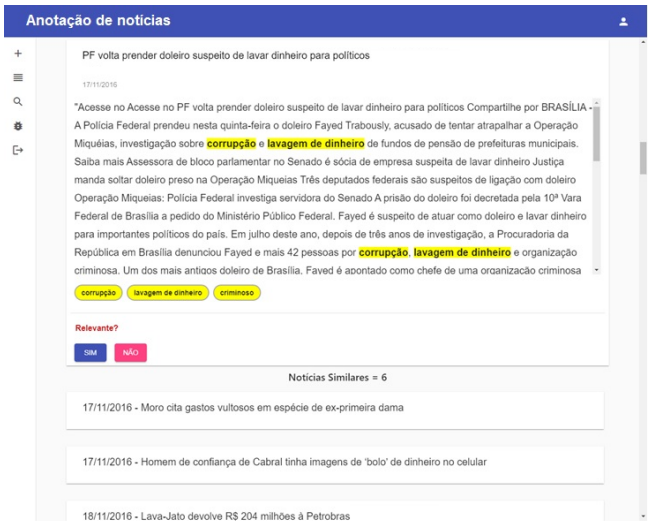

Figura IV.2: Formulário para anotação do corpus FraudeCorpusBR-M

Em seguida, os juízes receberam instruções com objetivo de padronizar o processo de anotação que continham informações para realização da anotação das notícias nas classes *relevante* e *não relevante*. A notícia é rotulada com a classe *relevante* caso existam no texto fatos que caracterizem que a empresa, parceiro ou empregado estejam envolvidos em algum escândalo de fraude ou corrupção. É apresentado a seguir um exemplo de trecho de notícia veiculada pelo Valor econômico com essas características:

> "Palocci teria cobrado pessoalmente valor relativo a sondas do pré-sal SÃO PAULO - Em seu depoimento ao Ministério Público Federal, o ex-executivo da Odebrecht Márcio Faria da Silva disse que o pedido de vantagem indevida realizado por Pedro Barusco, ex-gerente da Petrobras, foi realizado apenas ao fim da licitação para a construção de sondas de extração do pré-sal. O ex-ministro Antonio Palocci teria pessoalmente cobrado de Marcelo Odebrecht a propina. Em seu relato, Márcio Faria relembrou o processo, que começou em outubro de 2009. Na ocasião, a Petrobras lançou duas licitações que consistiam na construção de sete sondas de perfuração e duas DRUs

(Drilling Rig Unit, unidades de sonda de perfuração), que seriam utilizadas para águas ultraprofundas."

Já as notícias da classe *não relevante* são rotuladas quando não possuem as características mencionadas na classe anterior. É apresentado a seguir um exemplo de notícia dessa classe veiculada pelo Estadão *"Grupo rouba R$ 100 mi e leva cenário de guerra à fronteira Brasil-Paraguai Armados com fuzis automáticos e metralhadoras ponto 50, os criminosos bloquearam ruas, incendiaram veículos e dispararam rajadas contra prédios público"*. É possível notar que essa notícia trata-se de um crime de roubo efetuado por uma quadrilha e não é escopo da proposta do trabalho em questão.

Ao final desse processo, foram anotadas manualmente 181 notícias na classe *relevante* e 1480 notícias na classe *não relevante*, totalizando 1661 notícias. Esse conjunto de notícias foi identificado como FraudeCorpusBR-M. A figura IV.3 ilustra esse processo.

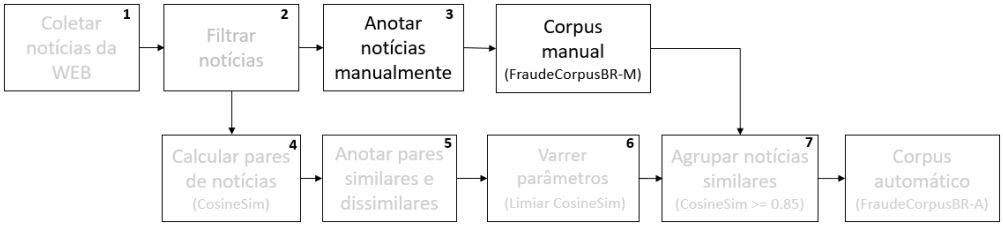

Figura IV.3: Etapas da construção do corpus FraudeCorpusBR-M

IV.1.2 FraudeCorpusBR-A

O processo de construção do corpus automático (FraudeCorpusBR-A) está ilustrado na figura IV.4 e as etapas seguidas para sua criação estão descritas nos próximos parágrafos desta seção. Esse processo foi responsável por aumentar a quantidade de notícias anotadas, conforme sugerido por Sinclair and Carter [2004]. Sendo assim, foi utilizada uma estratégia indireta de anotação que consiste em anotar automaticamente as notícias similares às que haviam sido anotadas no FraudeCorpusBR-M.

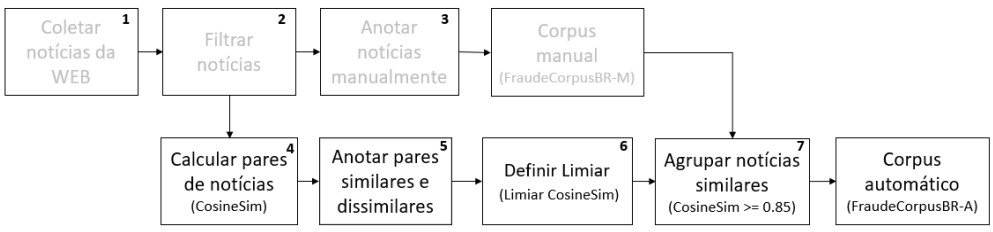

Figura IV.4: Etapas da construção do FraudeCorpusBR-A

Na etapa 4, foi utilizado o modelo de linguagem *Word2Vec* [Mikolov et al., 2013], pré treinado em português, para obter os *embeddings* (de 100 dimensões) de cada palavra do texto das notícias. Para obter os *embeddings*, foi desenvolvido um *script* em *Python* com auxílio da biblioteca *gensim*

em que cada embedding foi obtido de um vetor de embeddings gerado com base no modelo de linguagem *Word2Vec* pré treinado utilizando o corpus *wikipedia* em Português. Para cada texto, foi calculada a média dos *embeddings* de todas as palavras do texto, resultando assim em um *embedding* médio [Melamud et al., 2016] para cada notícia. Por fim, para cada par de notícias, foi calculada a similaridade do cosseno (CosineSim)[Elberrichi et al., 2008] dos *embeddings* médios.

As etapas 5 e 6 foram executadas com objetivo de definir um limiar de CosineSim que melhor separasse as notícias similares das não similares para que, a partir da anotação manual de uma notícia como *relevante* ou *não relevante*, suas similares fossem automaticamente anotadas. Sendo assim, a etapa 5 se encarregou de preparar os dados para etapa 6. Para isso, de posse dos pares de notícias e seus respectivos valores de CosineSim, foi efetuada a seleção de 76 pares de notícias, em que cada par foi anotado como *similar* ou *dissimilar*. O objetivo era avaliar qual o valor de CosineSim que melhor separava pares similares de pares não similares. Esse processo de anotação foi efetuado por três juízes, em que cada um anotou um subconjunto dos 76 pares de notícias.

Em seguida, a etapa 6, foi responsável por efetivamente definir o melhor limiar de ConsineSim para o processo de anotação automática. Para isso, foi efetuada uma varredura do parâmetro ConsineSim que utilizou os valores começando com 0.55. Incrementos de 0.05 foram efetuados até que a precisão da classe *similar* alcançasse 100%. O grau que obteve o melhor resultado foi o 0,85 de CosineSim. A escolha pela precisão de 100% na classe *similar* foi tomada pois o objetivo era anotar manualmente uma notícia como *relevante* ou *não relevante* e que suas similares fossem anotadas automaticamente a partir de um limiar seguro. Ou seja, ao anotar manualmente uma notícia com uma das classes (*relevante* ou *não relevante*), todas as notícias fora do corpus FraudeCorpusBR-M com grau de similaridade maior ou igual 0.85 eram também anotadas com o mesmo rótulo da notícia anotada manualmente. Outro ponto a ser observado é que a medida em que a precisão da classe *similar* aumentava, a precisão da classe *dissimilar* diminuia. Isso quer dizer que ao escolhermos o limiar de 0,85, 100% dos pares da classe *similar* eram classificados corretamente, porém somente 44% dos pares da classe *dissimilar* eram classificados corretamente. Apresentamos abaixo a tabela IV.1 que mostra os resultados obtidos na varredura de parâmetros.

Tabela IV.1: Varredura de parâmetros - Precisão - limiar CosineSim

CosineSim	Similar (Precisão)	Dissimilar(Precisão)
0.85	**100%**	**44%**
0.80	92%	51%
0.75	89%	59%
0.70	81%	66%
0.65	76%	81%
0.60	70%	89%
0.55	65%	100%

No passo 7, todas as notícias que possuíam um grau de similaridade maior ou igual ao liminar de 0.85 com qualquer uma das 1661 notícias anotadas manualmente foram automaticamente anotadas com o mesmo rótulo da notícia anotada manualmente. Desta forma foi criado um segundo corpus, contendo 20.717 notícias anotadas unicamente de forma automática, que foi chamado de FraudeCorpusBR-A. O total de notícias anotadas em cada corpus está apresentado na tabela IV.2.

Tabela IV.2: Notícias rotuladas

Título do rótulo	Notícias	Relevante	Não relevante
Manual	1.661	181	1.480
Automática	20.717	6.752	13.965
Total	22.378	6.933	15.445

IV.1.3 Análise exploratória do FraudeCorpusBR

A análise exploratória do FraudeCorpusBR foi realizada utilizando a linguagem de programação Python e as bibliotecas *pandas-profiling*, *pandas*, *wordcloud* e *matplotlib*. O objetivo dessa tarefa foi sintetizar os dados e aplicar técnicas de visualizações que permitissem evidenciar padrões não facilmente detectáveis nos dados originais. A tabela IV.3 nos permite observar estatísticas sobre o FraudeCorpusBR como a quantidade de dados faltantes dos atributos título, texto, origem, url, rótulo e os embeddings das notícias, a quantidade de atributos, a quantidade de exemplos, assim como o espaço necessário em memória para carregar o *dataset*. Já a tabela IV.4 mostra os tipos de atributos encontrados em que é possível observar a predominância do tipo numérico com 100 unidades. Além dos dados numéricos, o FraudeCorpusBR também possui mais 2 atributos categóricos, que não possuem valores quantitativos, mas, ao contrário, são definidas por várias categorias.

Tabela IV.3: Estatísticas do FraudeCorpusBR

Estatística	Valores
Quantidade de atributos	105
Quantidade de exemplos	22.378
Quantidade de dados faltantes	16.156
% Dados faltantes	0.6%
Espaço total em memória	309.1 MiB

Tabela IV.4: Tipos de atributos do FraudeCorpusBR

Tipo	Quantidade
Numéricos	100
Categóricos	5

Também foi gerada uma nuvem de palavras em que foi possível observar as palavras que apare-

cem com mais frequência nos títulos das notícias. Quanto maior é a palavra, mais vezes ela aparece no texto. Essa é uma ótima ferramenta para tentar entender tipos de termos estão influenciando mais nas classificações dos modelos.

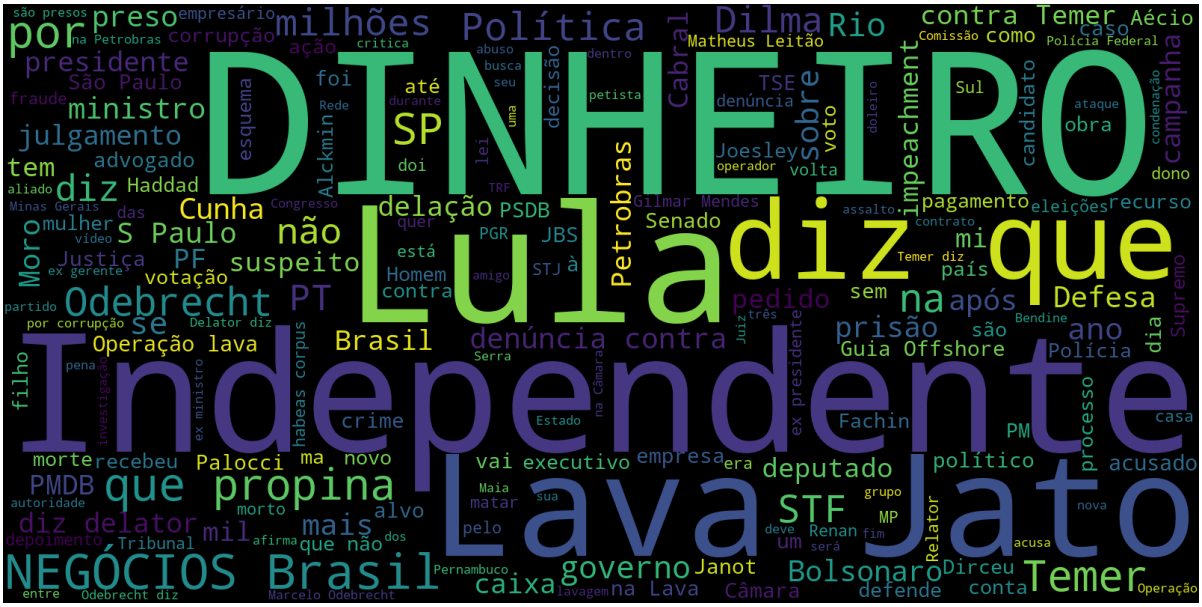

Figura IV.5: Nuvem de palavras do FraudeCorpusBR

Além da nuvem de palavras, foi gerado também um histograma que mostra o número de notícias por quantidade de palavras. Na figura IV.6 é possível observar uma concentração maior de notícias dentro de intervalo de 100 a 1000 palavras.

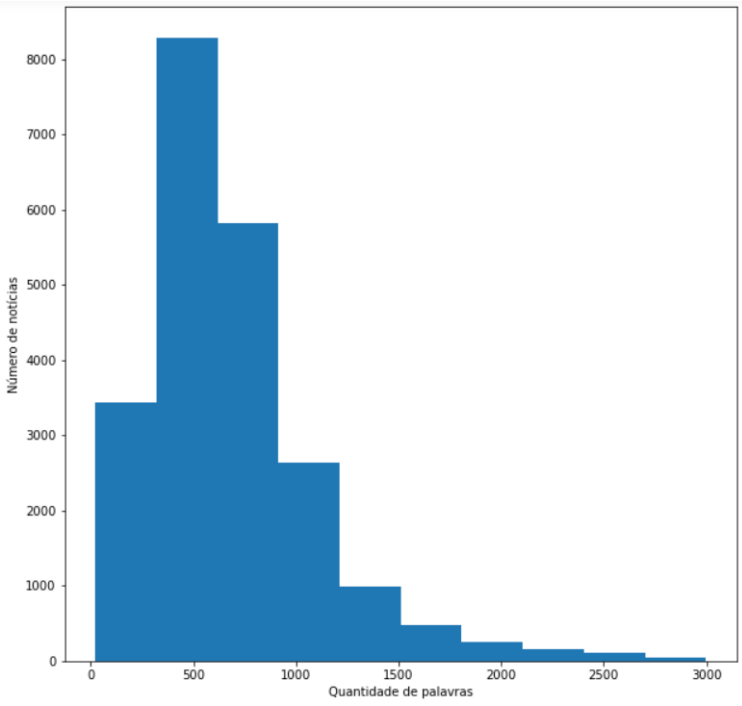

Figura IV.6: Histograma - Número de notícias x Quantidade de palavras

Capítulo V Metodologia

A metodologia apresentada neste capítulo envolve a criação de um modelo de classificação de notícias sobre fraude e corrupção. O objetivo é classificar cada notícia em *relevante* ou *não relevante* para instauração de processo investigativo. Esse modelo utilizou como base os conjuntos de dados FraudeCorpusBR-M e FraudeCorpusBR-A, cujos processos de criação foram descritos na seção IV. O processo de criação do modelo seguiu as etapas de engenharia de *feature*, escolha dos classificadores e treinamento dos seguintes classificadores: Rede Neural (NN), *Random Forest* (RF), *Suport Vector machine* (SVM) e Regressão Logística (RL). Por fim, para melhorar desempenho do modelo, foi criado um *ensemble* dos 4 classificadores escolhidos.

V.1 Classificadores

A escolha dos melhores classificadores foi baseada nas 4 melhores métricas de F1 da execução preliminar dos experimentos e contou com o treinamento e a avaliação de 8 classificadores distintos - Rede Neural (RN), *Random Forest*(RF), *Naive Bayes*(NB), KNN, *Support Vector Machine*(SVM), Árvore de decisão (AD), *AdaBoost* (AB) e Regressão Logistica (RL), que usaram como entrada as features geradas do conjunto de dados FraudeCorpusBR-A. O resultado da execução preliminar pode ser observado na tabela V.1. Os classificadores que obtiveram os 4 melhores resultados foram *Support Vector machine*(SVM), Rede Neural (NN), *Random Forest* (RF) e Regressão Logística (LR).

Tabela V.1: Resultados da execução preliminar - Escolha dos melhores classificadores

Ranking	Modelo	F1
1º	SVM	92,6%
2º	NN	92,2%
3º	RF	92,1%
4º	RL	91,9%
5º	KNN	91,8%
6º	AD	91,8%
7º	AB	89,6%
8º	NB	88,4%

V.2 Engenharia de *features*

Outra etapa importante para construção do modelo foi a engenharia de *feature*. O objetivo dessa etapa foi definir um conjunto de *features* que melhor representasse os dados e que fosse utilizado de forma mais eficiente pelo modelo. Foram geradas 102 novas *features* a partir do conjunto de dados FraudeCorpusBR. Das 102 *features* geradas, 100 representam as dimensões dos *embeddings* gerados para cada notícia (*Embeddings*), uma (1) representa a presença ou não de um dos FraudTerms no texto (FraudTerms) e uma (1) a presença ou nao de uma das MoneyRegex[1] no texto (MoneyRegex).

Para faciliar a organização das *features* que foram submetidas ao treinamento dos classificadores em cada um dos experimentos, foram criados 3 conjuntos de *features* denonimados *lists*, *emb* (*embeddings* de 100 dimensões) e *emb+lists* conforme figura V.1.

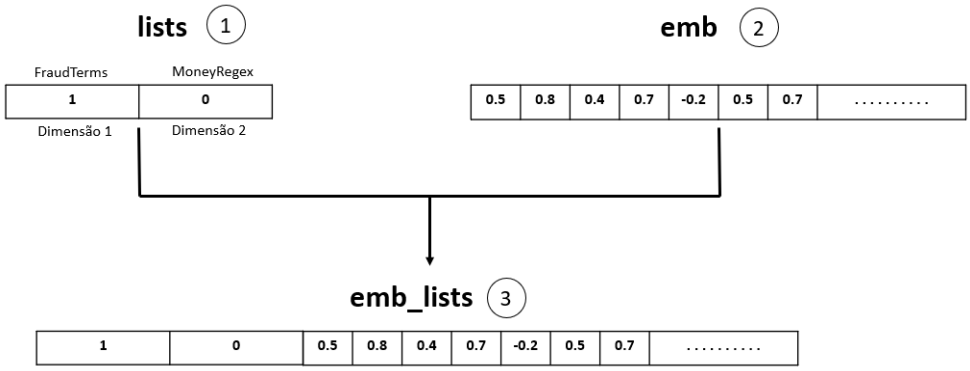

Figura V.1: *Features* do modelo de classificação

O primeiro conjunto de *feature* gerado foi o *lists*, composto de duas *features* (dimensões *one hot*). Seu processo de construção contou com a criação de uma lista (FraudTerms) e uma expressão regular (MoneyRegex) que serviram de base para definição de um vetor *one hot* de 2 dimensões (*lists*), em que cada dimensão possui o valor 0 ou 1, de acordo com as seguintes regras:

- Dimensão 1: valor 1 se o texto possui pelo menos um dos termos da lista FraudTerms, senão, 0.
- Dimensão 2: valor 1 se o texto possui pelo menos um termo que combina com expressão regular MoneyRegex, senão, 0 .

O segundo conjunto de *features* criado foi a *emb*, um vetor de *embeddings* médio do texto (*emb*) de 100 dimensões. Cada dimensão do *embedding* virou uma *feature*, totatilizando 100 novas *features*. O terceiro conjunto de *features* consiste em um vetor (*emb+lists*) composto por 102 *features*, resultante da junção do vetor de *embedding* (*emb*) com o vetor *one hot* de termos (*lists*).

[1] ${0,1}\d |(mil|(m)ilh(ares|ão|ões)).{0,3} (real|reais|euro|d[oó]lar)

No final da etapa de engenharia de *feature* foram gerados três conjuntos de *features* conforme conforme tabela V.2.

Tabela V.2: Conjuntos de Features

Conjunto	Features	Quantidade de features
lists	FraudTerms e MoneyRegex	2
emb	Embeddings	100
emb+lists	FraudTerms, MoneyRegex e Embeddings	102

V.3 Modelo de classificação de notícias

A criação do modelo de classificação de notícias está organizado em duas etapas. A etapa 1 iniciou após a escolha dos classificadores, descrita na seção V.1 e a execução da engenharia de features, descrita na seção V.2. Nessa etapa, os classificadores Rede Neural (NN), *Random Forest* (RF), *Support Vector machine*(SVM) e Regressão Logística (LR) foram treinados com os conjuntos de *features lists*, *emb* e *emb+list*, a fim de comparar seus desempenhos para a tarefa em questão. As configurações de cada classificador podem ser vistas abaixo:

- Rede Neural: *Multi Layer perceptron* (MLP) com *backpropagation*, 100 neurônios na primeira camada oculta e 50 neurônios na segunda. Função de ativação "ReLu", número máximo de iterações de 200, taxa de aprendizagem de 0,0001 e função de custo Adam (*stochastic gradient-based optimizer*).

- Regressão Logística: Função de regularização "Ridge L2". Parâmetro C (*Penalty Strength*) igual a 1,0.

- *Random forest*: Número de árvores igual a 50. Número mínimo de exemplos para divisão (*Minimum Samples Split*) igual a 5.

- *SVM*: Tipo SVM *Cost(C)* 1,00 e *regression lost episilon* 0,1. *Kernel RBF*. Parâmetro de otimização *Numerial tolerance* 0,001 *Iteration limit* 100.

Em seguida, a etapa 2 foi iniciada e contemplou a criação de um *ensemble* dos classificadores treinados na etapa 1. O objetivo do ensemble foi combinar as predições dos 4 classificadores para tentar criar um modelo com um desempenho melhor se comparado aos resultados individuais de cada um. Sendo assim, os resultados das classificações dos 4 modelos iniciais (RN, RF, SVM e RL) foram utilizados para criar um novo conjunto de dados denominado FraudeCorpusPredBR. Esse novo conjunto de dados contém o resultado das probabilidades das classes *relevante* ou *não relevante* e a classe predita de cada um dos modelos da etapa 1. A tabela V.3 representa uma amostra das predições realizadas por um dos classificadores da etapa 1. São apresentadas a probabilidade para

classe *relevante*, a probabilidade para classe *não relevante* e a classe predita para o classificador SVM.

SVM			
Notícia	Probabilidade relevante	Probabilidade não relevante	Classe predita
N1	0,67	0,33	relevante
N2	0,28	0,72	não relevante
N3	0,64	0,36	relevante
N4	0,28	0,72	não relevante
N5	0,29	0,71	não relevante
N6	0,31	0,69	não relevante
N7	0,23	0,77	não relevante

Tabela V.3: Exemplo ilustrador das probabilidade e predição de classe do classificador SVM

De posse do FraudeCorpusPredBR, foram criadas 12 novas features resultantes da probabilidade da classe *relevante*, da probabilidade da classe *não relevante* e da classe predita de cada um dos 4 classificadores. Esse segundo dataset foi submetido para treinamento de uma segunda camada de modelos, contendo 4 novos classificadores que utilizaram os mesmos algoritmos da primeira camada: Rede Neural, *Random Forest*, *Support Vector Machine* e Regressão Logística. No final da execução da segunda camada, os classificadores que obtiveram os melhores resultados foram o SVM e a Regressão logística. Na figura V.2 é possível observar as etapas de criação do modelo de classificação.

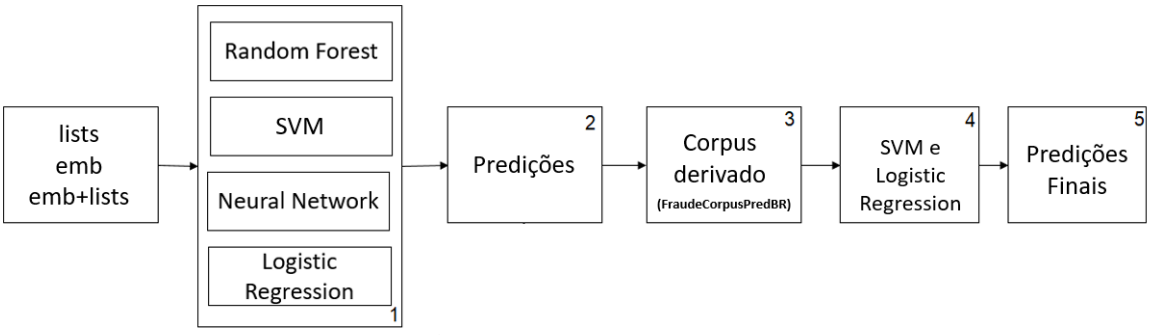

Figura V.2: Etapas da construção do modelo de classificação de notícias

Capítulo VI Experimentos

Os experimentos foram realizados em quatro etapas, a primeira etapa considerou a avaliação dos quatro classificadores com um conjunto de *features* gerado a partir do corpus anotado manualmente por um grupo de especialistas (FraudeCorpusBR-M); a segunda etapa considerou o treinamento utilizando um conjunto de *features* gerado com base em um corpus anotado automaticamente (FraudeCorpusBR-A); a terceira etapa considerou os mesmos classificadores e corpus da segunda etapa, mas foi efetuado um balanceamento das classes *relevante* e *não relevante*; e por último, a avaliação foi feita considerando um conjunto de *features* gerado a partir de um conjunto de dados derivado (FraudeCorpusPredBR), derivado das predições do melhor modelo escolhido com base na execução dos primeiros experimentos. Esse conjunto de dados foi utilizado para o treinamento da segunda camada de modelos ensemble visando melhoria dos resultados obtidos pelos modelos individualmente.

VI.1 Experimento 1 - FraudCorpus-M

Como primeiro experimento, cada um dos 4 classificadores (RN, RF, RL e SVM) foi treinado, individualmente, com as *features emb* e *emb+lists*, utilizando o FraudeCorpusBR-M. Para selecionar os conjuntos de treino e teste, foi utilizada a técnica de *Cross Validation* estratificado com um número de *folds* igual a 5. Foi utilizado também como *baseline*, as features *one-hot encoder* dos textos das notícias junto com a feature *lists*, visando demonstrar a contribuição das features emb e emb+lists. Podemos observar o resultado desta primeira etapa de treinamento na Tabela VI.1, com destaque para o modelo SVM que obteve um *F1 Score* de 91,6% ao ser treinado utilizando a *feature emb+lists*.

Tabela VI.1: Resultados dos modelos individuais - FraudeCorpusBR-M

Modelo-Features	F1	AUC	CA	Precision	Recall
RN-baseline	82,5%	83,7%	83,2%	82,8%	83,2%
RF-baseline	81,6%	87,9%	82,4%	81,9%	82,4%
RL-baseline	83,5%	89,8%	83,7%	83,4%	83,7%
SVM-baseline	28,3%	63,1%	35,9%	65,0%	35,9%
RN-emb	90,5%	93,6%	90,9%	90,2%	90,9%
RF-emb	91,0%	92,8%	91,7%	90,8%	91,7%
RL-emb	90,1%	93,0%	91,2%	90,1%	91,2%
SVM-emb	90,9%	93,4%	91,5%	90,7%	91,5%
RN-emb+lists	91,5%	94,2%	91,8%	91,3%	91,8%
RF-emb+lists	91,3%	92,0%	92,1%	91,2%	92,1%
RL-emb+lists	90,5%	93,7%	91,4%	90,4%	91,4%
SVM-emb+lists	**91,6%**	**93,3%**	**92,0%**	**91,4%**	**92,1%**

VI.2 Experimento 2 - FraudCorpus-A

No segundo experimento, os 4 classificadores foram novamente treinados individualmente com as mesmas *features* do primeiro experimento, mas, desta vez, utilizando o FraudeCorpusBR-A. Para selecionar os conjuntos de treino e teste, também utilizamos *Cross Validation* estratificado com um número de *folds* igual a 5. Os resultados desse segundo experimento estão exibidos na Tabela VI.2, com destaque para o modelo RN que obteve um *F1 Score* de 92,6% ao ser treinado utilizando a *feature emb+list*.

Tabela VI.2: Resultados dos modelos individuais - FraudeCorpusBR-A

Modelo-Features	F1	AUC	CA	Precision	Recall
RN-emb	92,5%	97,8%	92,5%	92,5%	92,5%
RF-emb	92,2%	97,2%	92,2%	92,2%	92,2%
RL-emb	91,0%	96,6%	91,1%	91,0%	91,1%
SVM-emb	78,1%	82,4%	77,6%	79,7%	77,6%
RN-emb+lists	**92,6%**	**97,8%**	**92,6%**	**92,6%**	**92,6%**
RF-emb+lists	92,1%	97,2%	92,1%	92,1%	92,1%
RL-emb+lists	91,0%	96,6%	91,0%	91,0%	91,0%
SVM-emb+lists	82,0%	95,6%	82,0%	82,6%	81,7%

Podemos observar, analisando as Tabelas VI.1 e VI.2 que o modelo SVM obteve o melhor desempenho ao ser treinado com a *feature* emb+lists no FraudeCorpusBR-M (F1 = 91,6%) e o modelo RN obteve o melhor desempenho ao ser treinado com a *feature emb+lists* no FraudeCorpusBR-A (F1 = 92,6%).

VI.3 Experimento 3 - FraudCorpusPredBR

Baseado nos resultados dos primeiros experimentos, o FraudeCorpusBR-A foi escolhido por obter os melhores resultados junto com conjunto de *feature emb+lists*. O FraudeCorpusBR-M (1661 exemplos) escolhido para ser submetido aos modelos treinados para realização das predições que serviram como base de criação do *dataset* derivado. Cada um dos 4 classificadores gerou como saída 3 variáveis (probabilidade para o rótulo relevante, probabilidade para o rótulo não relevante e o rótulo predito). Dessa forma, temos como saída dessa etapa de predição um conjunto de dados composto de 12 colunas e 1661 linhas, formando o *dataset* denominado de FraudeCorpusPredBR.

Para a realização do próximo experimento, o FraudeCorpusPredBR foi utilizado para o treinamento, onde foi submetido a segunda camada de modelos (ensemble dos classificadores), que usou os resultados das predições dos modelos anteriores como dados de treinamento. Os mesmos 4 classificadores (RN, RF, RL e SVM) foram utilizados, mas desta vez, foram usadas doze (12) novas *features* de entrada, um vetor de 8 dimensões (2 probabilidades para cada um dos 4 modelos) e 4 variáveis categóricas (os 4 rótulos preditos pelos 4 modelos). Os treinamentos desse experimento foram feitos utilizando o FraudCorpusPredBR, que foi dividido em 65% para treino (1080 exemplos) e 35% para teste (581 exemplos).

Na Tabela VI.3 podemos observar o resultado do terceiro experimento, onde é possível notar uma melhora nos resultados da classificação, passando de um F1 de 92,6, no melhor caso do segundo experimento (conforme tabela VI.3 na seção VI.2), para um F1 de 93,4 dos modelos RL e SVM no terceiro experimento.

Tabela VI.3: Resultados dos modelos individuais com corpus derivado

Modelo	F1	AUC	CA	Precision	Recall
RN-Ensemble-emb+lists	92,8%	95,2%	93,5%	92,9%	93,5%
RL-Ensemble-emb+lists	**93,4%**	**96,2%**	**94,0%**	**93,6%**	**94,0%**
RF-Ensemble-emb+lists	93,0%	95,5%	93,6%	93,2%	93,6%
SVM-Ensemble-emb+lists	**93,4%**	**95,6%**	**94,0%**	**93,6%**	**94,0%**

VI.4 Experimento 4 - Balanceamento do FraudeCorpus-A

Embora o experimento 3 tenha produzido um bom resultado, foi necessário verificar se o desbalanceamento entre classes do FraudCorpusBR-A estava gerando distorções na classificação da classe minoritária (Relevante). Com isso, efetuamos o balanceamento do FraudeCorpusBR-A com a técnica de *Under-sampling* conforme proposto por Dal Pozzolo et al. [2015]. A tabela VI.4 mostra o *dataset* antes e depois do balanceamento.

Tabela VI.4: FraudeCorpusBR-A - Desbalanceado x Balanceado

Balanceado	Notícias	Relevante	%Relevante	Não relevante	%Não relevante
Não	20.717	6.752	32,59%	13.965	67,41%
Sim	13.416	6.708	50,00%	6.708	50,00%

Em seguida, o FraudeCorpusBR-A balanceado foi submetido ao treinamento dos 4 classificadores o que permitiu observar que houve uma piora nas métricas de precisão de classe relevante. A precisão da classe *relevante* caiu de 62,4% com o *dataset* desbalanceado para 60,8% com *dataset* balanceado. Ou seja, o desbalanceamento do *dataset* não está gerando distorções na classificação do modelo para a classe relevante. Comparamos as matrizes de confusão para as classes dos *datasets* FraudeCorpus-A balanceado e FraudeCorpus-A (Desbalanceado). Os resultados desse experimento estão exibidos na figura VI.1.

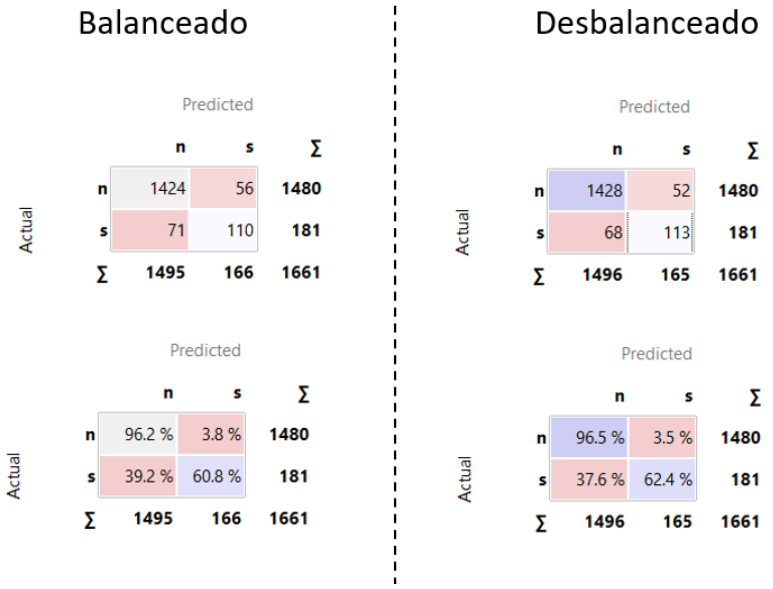

Figura VI.1: Matriz de confusão - Balanceado x Desbalanceado

Desta forma, os classificadores RL e SVM foram escolhidos para serem os classificadores da última camada, ficando a arquitetura geral do modelo proposto conforme ilustrada na Figura V.2.

Capítulo VII Conclusões

Esta pesquisa apresentou uma proposta para automatizar a tarefa de monitoramento e classificação de notícias em Português sobre fraude e corrupção conforme a relevância para abertura de processo investigativo interno. O trabalho apresentou exemplos que constataram que nem todas as notícias de fraude e corrupção são relevantes para abertura de processo investigativo interno e muitas vezes essa tarefa se torna muito custosa se executada de forma manual. Para tentar resolver esse problema, foi proposto um modelo de classificação que combina a predição de vários classificadores (*Ensemble*). Esse modelo é capaz de classificar textos de notícias em Português sobre fraude e corrupção como relevantes ou não para instauração de processo investigativo interno e obteve o *F1 Score* de 93,4%.

VII.1 Contribuições

Foram apresentadas algumas contribuições ao longo da presente pesquisa que estão relacionadas a seguir:

- FraudeCorpusBR: Foi construído um corpus que possui notícias em Português sobre fraude e corrupção anotadas com o rótulo *relevante* ou *não relevante* para instauração de processo investigativo. Para gerar esse corpus foram criados 46 *Web crawlers* de veículos de notícias do Brasil que foram responsáveis por coletar os textos que serviram de base para treinamento do modelo.

- Corpus anotado automaticamente (FraudCorpusBR-A): Outra contribuição do trabalho foi a construção de um corpus FraudCorpusBR-A anotado automaticamente a partir do corpus rotulado manualmente (fraudCorpusBR-M). Foi utilizado o modelo de *word2vec* para representação vetorial dos textos. Com isso foi possível calcular a similaridade do cosseno para agrupar as notícias por similaridade. A cada notícia anotada manualmente, suas similares eram anotadas automaticamente. Durante os experimentos foi demonstrado que os modelos que usaram um corpus anotado automaticamente obtiveram melhores resultados que aqueles treinados exclusivamente no corpus anotado manualmente.

- Modelo de classificação: A principal contribuição na presente pesquisa foi a criação de um

modelo de classificação de notícias sobre fraude e corrupção capaz de sugerir a instauração de processo investigativo. Para criação do modelo, foi executada uma sequência de tarefas que começou com escolha dos melhores classificadores para atividade proposta nesta dissertação. Foram testados os classificadores Rede Neural (RN), *Random Forest*(RF), *Naive Bayes*(NB), KNN, *Support Vector Machine*(SVM), Árvore de decisão (AD), *AdaBoost* (AB) e Regressão Logística (RL) e os 4 que tiveram as melhores métricas de F1 foram os classificadores SVM, NN, RF e RL. Depois de escolhidos os classificadores, foram executadas rodadas de experimentos para decidir qual era o melhor modelo para a tarefa proposta nessa dissertação. O modelo RN obteve o melhor desempenho ao ser treinado com a *feature emb+list* no FraudeCorpusBR-A (F1 = 92,6%) e superou o modelo SVM que obteve o melhor desempenho ao ser treinado com a *feature emb+lists* no FraudeCorpusBR-M (F1 = 91,6%). Foi possível observar também uma melhora nos resultados obtidos ao adotar o *ensemble* de classificadores na segunda camada, passando de um F1 de 92,6%, no melhor caso do segundo experimento, para um F1 de 93,4% dos modelos RL e SVM no terceiro experimento.

VII.2 Aspectos relevantes

- Outro aspecto importante observado nesse trabalho foi o desbalanceamento entre classes no FraudeCorpusBR. Apresentamos uma preocupação pelo fato de modelos desbalanceados gerarem distorções quando classificam exemplos da classe minoritária. Com objetivo de identificar se essa distorção estava ocorrendo no modelo proposto nesse trabalho, efetuamos um último experimento, em que o *dataset* FraudeCorpusBR-A foi balanceado utilizando a técnica de *Under-sampling*. Os exemplos da classe majoritária foram reduzidos até que a quantidade de exemplos das classes minoritária e majoritária se igualassem. Em seguida, o FraudeCorpusBR-A balanceado foi submetido ao treinamento e foi possível observar uma piora na métrica de precisão da classe minoritária se comparado ao *dataset* FraudeCorpusBR-A desbalanceado. Sendo assim, podemos concluir que o *dataset* FraudeCorpusBR-A desbalanceado não apresentou distorções na tarefa de classificação, e portanto foi decidido não utilizar o *dataset* balanceado.

- Não foram encontrados na literatura trabalhos cujos objetivos são a classificação de notícias de fraude e corrupção para instauração de processo investigativo. Essa pesquisa poderá servir de ponto de partida para o desenvolvimento de trabalhos futuros que venham apresentar propostas de evolução de modelos nesta tarefa.

VII.3 Trabalhos futuros

Foram identificadas oportunidades de melhoria que estão apresentadas a seguir como trabalhos futuros:

- Utilizar modelos com a arquitetura *Transformer*: é recomendado que sejam testados modelos de linguagem baseados na arquitetura *Transformer*, pois esses modelos permitem criar representações vetoriais (*embeddings*) com uma qualidade superior. É esperado que ao utilizar modelo de linguagem com essa arquitetura, o agrupamento de notícias por similaridade seja feito com mais precisão [Wolf et al., 2020].

- Utilização do modelo *Doc2vec*: A estratégia escolhida para criar a representação vetorial da notícia foi utilizar a média dos *embeddings* das palavras presentes no texto da notícia. Uma sugestão para trabalhos futuros seria criar essa representação utilizando modelos como o *Doc2vec* que já gera o *embedding* para a sentença. É recomendado efetuar comparação dos resultados dos modelos com essas duas estratégias [Le and Mikolov, 2014].

Referências Bibliográficas

Amudha, S. and Phil, M. (2017). Web crawler for mining web data. *International Research Journal of Engineering and Technology*, 3:128–136. 5

Bolton, R. J. and Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical science*, 17(3):235–255. 1

Calomiris, C. W. and Mamaysky, H. (2019). How news and its context drive risk and returns around the world. *Journal of Financial Economics*, 133(2):299–336. 15

Costa, A. P. P. d. and Wood Jr, T. (2012). Fraudes corporativas. *Revista de Administração de Empresas*, 52:464–472. 1

Dal Pozzolo, A., Caelen, O., Johnson, R. A., and Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE symposium series on computational intelligence*, pages 159–166. IEEE. 11, 29

de Brito, J. G. P. P. (2018). Deteção de fraude em redes financeiras com modelação baseada em agentes. 2

Elberrichi, Z., Rahmoun, A., and Bentaalah, M. A. (2008). Using wordnet for text categorization. *International Arab Journal of Information Technology (IAJIT)*, 5(1). 7, 20

Gottschalk, P. (2016). Fraud examiners in white-collar crime investigations. In *Research Handbook on Corporate Social Responsibility in Context*. Edward Elgar Publishing. 1

Gürcan, F. (2018). Multi-class classification of turkish texts with machine learning algorithms. In *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–5. IEEE. 15

Hallac, I. R., Ay, B., and Aydin, G. (2018). Experiments on fine tuning deep learning models with news data for tweet classification. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pages 1–5. IEEE. 15

Han, J., Pei, J., and Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann. 10, 11, 12

Junardi, W. and Khodra, M. L. (2020). Automatic multi-label classification for gdp economic-phenomenon news. In *2020 International Conference on ICT for Smart Society (ICISS)*, pages 1–6. IEEE. 14

Kumar, S., Sharma, A., Reddy, B. K., Sachan, S., Jain, V., and Singh, J. (2021). An intelligent model based on integrated inverse document frequency and multinomial naive bayes for current affairs news categorisation. *International Journal of System Assurance Engineering and Management*, pages 1–15. 14

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR. 33

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer. 8

Li-Juan, Z., Feng, Z., Qing-Qing, P., Xin, Y., and Zheng-Tao, Y. (2015). A classification method of vietnamese news events based on maximum entropy model. In *2015 34th Chinese Control Conference (CCC)*, pages 3981–3986. IEEE. 16

Lima, M., Silva, R., de Souza Mendes, F. L., de Carvalho, L. R., Araujo, A., and de Barros Vidal, F. (2020). Inferring about fraudulent collusion risk on brazilian public works contracts in official texts using a bi-lstm approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1580–1588. 14

Liu, H., Tao, L., and Liu, M. (2020). Research on financial fraud identification of listed companies based on text data mining. In *2020 International Conference on Image, Video Processing and Artificial Intelligence*, volume 11584, page 115841X. International Society for Optics and Photonics. 2, 14

Livingston, F. (2005). Implementation of breiman's random forest machine learning algorithm. *ECE591Q Machine Learning Journal Paper*, pages 1–13. 7

Mammone, A., Turchi, M., and Cristianini, N. (2009). Support vector machines. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(3):283–289. 7

Melamud, O., McClosky, D., Patwardhan, S., and Bansal, M. (2016). The role of context types and dimensionality in learning word embeddings. *arXiv preprint arXiv:1601.00893*. 20

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. , 6, 19

Opitz, D. and Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198. 9, 10

Palei, S. K. and Das, S. K. (2009). Logistic regression model for prediction of roof fall risks in bord and pillar workings in coal mines: An approach. *Safety science*, 47(1):88–96. 8

Quirk, P. J. (1997). Money laundering: muddying the macroeconomy. *Finance & Development*, 34(001). 1

Rasekh, I. (2015). A new competitive intelligence-based strategy for web page search. In *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pages 120–126. IEEE. 2

Simons, H. (2009). Neural networks and learning machines. *Hamilton, ON, Canada: Pearson Education, McMaster Univ.* 9

Sinclair, J. and Carter, R. (2004). *Trust the text: Language, corpus and discourse*. Routledge. 19

Sundaramoorthy, K., Durga, R., and Nagadarshini, S. (2017). Newsone—an aggregation system for news using web scraping method. In *2017 International Conference on Technical Advancements in Computers and Communications (ICTACC)*, pages 136–140. IEEE. , 5

Tan, B., Foo, S., and Hui, S. C. (2002). Web information monitoring for competitive intelligence. *Cybernetics &Systems*, 33(3):225–251. 4

Thaipisutikul, T., Tuarob, S., Pongpaichet, S., Amornvatcharapong, A., and Shih, T. K. (2021). Automated classification of criminal and violent activities in thailand from online news articles. In *2021 13th International Conference on Knowledge and Smart Technology (KST)*, pages 170–175. IEEE. 2, 13

Weichselbraun, A., Hörler, S., Hauser, C., and Havelka, A. (2020). Classifying news media coverage for corruption risks management with deep learning and web intelligence. In *Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics*, pages 54–62. 2, 13

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45. 33