Capítulo 2

Introdução à Análise de Sentimentos com Word Clouds

André Viana Tardelli, Angélica Fonseca da Silva Dias, Juliana Baptista dos Santos França

Abstract

The goal of this short course is to allow an introductory discussion about Data Science, specially the Sentiment Analysis area, in order to present its results graphically and statistically. The course introduces the basic concepts of natural language processing via text and argues for strategies for analyzing large amounts of data in order to detect general opinions about a given number of assessments provided. The construction of this analysis is made through the execution of structured stages, which determine each analysis step separately, such as Training, Optimization and Graph Generation.

Resumo

O objetivo deste capítulo é permitir uma discussão introdutória sobre a área de conhecimento de Ciência de Dados, e abordando a Análise de Sentimento, de maneira a apresentar seus resultados de forma gráfica e estatística. O curso introduz os conceitos básicos de processamento da linguagem natural via texto, e argumenta sobre estratégias de análise de grandes quantidades de dados a fim de detectar opiniões gerais sobre determinado número de avaliações fornecidas. A construção desta análise é feita via a execução de estágios estruturados, que determinam cada etapa da análise separadamente, como Treinamento, Otimização e Geração dos Gráficos.

2.1. Introdução

Devido à grande quantidade de produtos e serviços disponibilizados via meios digitais na sociedade atual, a opinião das pessoas que os utilizam é um fator cada vez mais valioso para o discernimento de sua qualidade [Zhang et al., 2012]. Dessa forma, diversos profissionais em empresas são responsáveis por processar e analisar o *feedback* dado pelos usuários em canais digitais para prover sugestões e ideias para melhorar os produtos disponibilizados pelas empresas.

Todavia, muitas vezes esse processamento e análise do *feedback* dos usuários por humanos acabam tornando-se ineficaz, visto que existe um número demasiadamente grande de comentários e avaliações a serem lidos. Dessa forma, diversos *feedbacks* acabam sendo perdidos, visto que é muito difícil gerar um compilado eficaz de informações capaz de mostrar a opinião geral que determinados usuários estão sentindo em relação a algum produto. Assim, notou-se a escassez de algum método que fosse capaz de destacar os *feedbacks* mais relevantes, de acordo com o tipo de opinião fornecida [Agarwal et al., 2015].

Ao longo dos últimos anos foi introduzido o conceito de análise de sentimento, que possibilitou explicitar opiniões de maneira mais direta, categorizando os elementos principais de uma massa de texto de acordo com a sua relevância tanto positivamente quanto negativamente. A identificação de sentimentos em artefatos textuais é uma das áreas de pesquisa mais destacadas em Processamento de Linguagem Natural desde o início dos anos 2000 [Liu, 2010]. Dessa maneira, o principal objetivo da análise de sentimentos é definir técnicas automáticas capazes de extrair informações subjetivas de textos em linguagem natural, como opiniões e sentimentos, a fim de criar conhecimento estruturado [Benevenuto et al., 2015]. Empresas relevantes no mercado podem analisar seus produtos e serviços de maneira geral e assim, obter um *feedback* mais preciso sobre a massa de comentários disponibilizados por seus usuários.

Este capítulo tem como foco mostrar os conceitos básicos para gerar uma análise de sentimento com base em uma massa de dados (grande volume de comentários), de modo a conseguir representar de maneira gráfica e interativa os resultados obtidos. Para isto, serão mostrados os conceitos de aprendizado de máquina e processamento de linguagem natural através da linguagem *Python*, em um ambiente colaborativo e fácil de ser preparado. Ao final, espera-se que os participantes tenham internalizado conceitos relacionados à: i) Aprendizado de Máquina ii) Processamento de Linguagem Natural iii) Análise de Sentimento e iv) Manipulação de *datasets* e geração de gráficos na linguagem *Python*. É importante destacar que será apresentado no texto a mineração (análise de sentimentos) usando métodos de classificação por aprendizado de máquina supervisionado. No entanto, esta mineração poderia ser conduzida por outras técnicas como as não supervisionadas (dicionários léxicos), ou semi-supervisionados.

O capítulo 2 está organizado de maneira estruturada, sendo constituído de uma breve compreensão das abordagens teóricas e práticas. A seção 2.2 contém a base do referencial teórico a ser discutido, a seção 2.3 possui as especificações de ambiente e bibliotecas a serem utilizadas, a seção 2.4 discute todos os passos a serem realizados durante a parte prática do capítulo, e a seção 2.5 contém uma breve discussão dos resultados alcançados.

2.2. Referencial Teórico

Antes de realizar uma análise de sentimento, é necessário rever os conceitos base que compõem a sua estrutura. Nesta seção, serão apresentadas as bases teóricas da área de Ciência de Dados que serão aplicadas em diversas partes da construção da análise a ser realizada neste capítulo.

2.2.1. Aprendizado de Máquina

Com o avanço da tecnologia, tornou-se cada vez mais necessário a criação de sistemas inteligentes capazes de simular ações realizadas por seres humanos. Essas simulações ocorrem através da aplicação de algoritmos de Aprendizado de Máquina, com o objetivo de extrair informações de dados fornecidos e consequentemente desenvolver um modelo geral que seja capaz de representar o problema estudado [Horta, 2015]. Dessa forma, diferentes técnicas na área de Ciência de Dados foram implementadas de maneira a induzir a tomada de alguma ação, seja baseada em experiências anteriores ou a partir de medidas de qualidade de respostas obtidas.

Segundo Monard e Baranauskas (2003), o conceito de Aprendizado de Máquina possui como objetivo desenvolver técnicas computacionais para adquirir conhecimento de maneira automática, de forma a possibilitar que a máquina possa tomar decisões autônomas. Esse paradigma tornou-se cada vez mais popular ao longo dos anos, visto que a implementação de processos indutivos possibilitou a criação de classificadores automáticos dado um conjunto pré-determinado de dados.

Dessa forma, a utilização de aprendizado de máquina trouxe diversas vantagens na área de processamento de texto, visto que as acurácias das classificações são comparáveis a de um ser humano, evitando assim a necessidade da intervenção de especialistas da área para analisar e gerar os classificadores principais de diferentes tipos de categorias [Sebastiani, 2002].

Na área de Aprendizado de Máquina, é possível treinar o seu algoritmo através de uma maneira supervisionada, não supervisionada ou semi supervisionada. Com o foco na análise de sentimentos, e de acordo com Benevenuto, Ribeiro e Araújo (2015), a supervisionada é embasada nos conceitos de aprendizagem de máquina partindo da definição de características que permitam distinguir entre sentenças com diferentes sentimentos, treinamento de um modelo com sentenças previamente rotuladas e utilização do modelo de forma que ele seja capaz de identificar o sentimento em sentenças até então desconhecidas. Já a não supervisionada não conta com treinamento de modelos de aprendizado de máquina e, em geral, são baseadas em tratamentos léxicos de sentimentos que envolvem o cálculo da polaridade de um texto a partir de orientação semântica das suas palavras. Por fim a semi supervisionada trata-se de uma grande oportunidade para quem não pode bancar o preço de treinar todos os seus dados. Este método permite-nos melhorar significativamente a acurácia, pois permite utilizar dados não treinados com uma pequena quantidade de dados treinados. Neste capítulo, será utilizado o aprendizado supervisionado para realizar a categorização dos elementos, sendo discutido com mais detalhes na seção 2.2.1.1.

2.2.2. Aprendizado Supervisionado

Uma abordagem muito comum para gerar um aprendizado supervisionado é a separação da massa de dados em uma massa de treino e uma massa de teste, contendo as informações principais a serem analisadas. Dessa forma, a massa de treino é responsável por possuir todas as informações a serem induzidas pelo algoritmo, enquanto a massa de teste será utilizada para averiguar se as instâncias principais daquele conjunto de dados conseguem ser corretamente previstos.

De acordo com Benevenuto et. al., (2015) o termo supervisionado é apresentado justamente pelo fato de exigir uma etapa de treinamento de um modelo com amostras previamente classificadas. O procedimento para realizar a aprendizagem de máquina compreende as seguintes etapas principais: I. obtenção de dados rotulados para uso em treino e para teste; II. Definição das *features* ou características que permitam a distinção entre os dados; III. Treinamento de um modelo computacional com um algoritmo de aprendizagem; e IV. Aplicação do modelo. Essas etapas serão conduzidas neste capítulo através da biblioteca SKLearn apresentada nas sessões abaixo deste capítulo.

2.2.3. Processamento de Linguagem Natural

Uma das grandes dificuldades na área de Ciência de Dados é a capacidade de fazer com que máquinas consigam processar e interpretar textos. Para conseguir simular essa interpretação, foram elaborados conjuntos de técnicas para representar elementos textuais de maneira computacional de modo a alcançar um nível de interpretação linguística em uma máquina a nível de um ser humano.

Segundo Liddy (2001), o conceito de Processamento de Linguagem Natural é definido por um conjunto de técnicas computacionais para analisar e representar naturalmente textos em um ou mais níveis linguísticos de análise com o propósito de alcançar um nível humano de interpretação para diversos tipos de tarefas ou aplicações. Este nível de interpretação pode ser definido em diversas categorias, se baseando em estruturas morfológicas, fonéticas, lexicais ou semânticas. Desta forma, diferentes abordagens podem ser feitas para alcançar um nível de interpretação desejada, de acordo com as necessidades de cada análise.

Como o objetivo do capítulo visa categorizar palavras-chave baseadas no sentimento geral, uma análise léxica será utilizada para processar estes dados de maneira a gerar uma análise individual de cada termo. Todavia, os textos a serem analisados se encontram muito simples, dificultando a relevância de termos de maior importância. Dessa forma, será necessário realizar um pré-processamento no corpus textual a ser analisado de maneira a otimizar a interpretação de todos os termos presentes no mesmo. Estes métodos de pré-processamento serão discutidos mais à frente na seção 2.4.

2.2.4. Análise de Sentimentos

A área de Análise de Sentimento tem como foco principal explicitar termos principais referentes a uma opinião em forma de texto [Serranoguerrero, 2015]. Dessa forma, é possível analisar uma extensa variedade de dados referentes a algum produto ou serviço,

possibilitando uma visão generalizada da opinião dos clientes da mesma através de relatórios ou recursos gráficos.

Todavia, antes de adentrar num detalhamento sobre seus métodos, é necessário especificar o escopo a ser trabalhado ao citar a temática de sentimento. Sendo um conceito estudado em diversas áreas como psicologia, computação e biologia, o sentimento é representado em diversos estudos da literatura de maneiras distintas [Ceci et. al., 2017].

Sentimento ou emoção indica uma carga de sentido específico presente em uma mensagem, que pode ser: raiva, surpresa, felicidade, alegria, tristeza, etc. Alguns métodos apresentam abordagens capazes de identificar qual sentimento uma sentença representa. Um exemplo clássico trata-se da abordagem léxica Emolex [Mohammad and Turney, 2013], a qual é baseada a partir da avaliação de milhares de sentenças em inglês para 9 sentimentos diferentes: *joy, sadness, anger, fear, trust, disgust, surprise, anticipation, positive, negative*. A força do sentimento representa a sua intensidade. Normalmente, este resultado é flutuante entre (-1 e 1). Há trabalhos que por exemplo medem a força de sentimentos nos títulos das notícias como o Magnet News[Reis et al., 2014] [Reis et al., 2015b], capaz de separar eficientemente para o usuário notícias boas de notícias ruins.

Partindo de uma avaliação cognitiva do assunto, Jung (2003) qualifica os sentimentos através de um sinal positivo ou negativo. Dessa forma, é possível obter a categorização de um sentimento em um viés computacional através da polarização fornecida em um termo ou conteúdo. Assim, é muito comum ver a área de análise de sentimento associada com temas como Processamento de Linguagem Natural, por exemplo.

A área de Análise de Sentimento muitas vezes é igualada a área de Mineração de Opinião, pois ambas surgiram com o intuito de realizar tarefas de identificação, classificação, análise de opiniões e sentimentos. Dessa forma, ambas podem ser classificadas como a área da computação que estuda todos os sentimentos, opiniões e emoções expressas em um texto [Ceci et. al., 2017].

2.3. Ferramentas Utilizadas

Nesta etapa, serão discutidas as especificações do ambiente, bibliotecas e ferramentas a serem importadas e utilizadas para gerar a análise. As ferramentas foram escolhidas tendo em mente a utilização da linguagem *Python*, devido a vasta quantidade de bibliotecas e *plugins* disponíveis para a realização da análise de dados e por ser uma linguagem relativamente simples para a realização dos trechos de código a serem executados. Todavia, não é necessário um extenso conhecimento da linguagem em si, visto que todas as estruturas e nomenclaturas básicas de função serão discutidas passo a passo ao longo do capítulo.

O ambiente a ser utilizado para a execução dos *scripts* em *Python* será a plataforma *Google Colaboratory*, que proporciona um ambiente online sem necessidade

de configurações complexas e fornece um processamento de dados sobre uma máquina virtual disponibilizada gratuitamente através de uma conta *Google.* Dessa forma, o único recurso necessário para todas as máquinas a serem utilizadas será a disponibilidade de computadores com acesso à internet.

Além disso, alguns *scripts* necessitaram de funcionalidades que requerem a importação de alguma biblioteca, que serão explicitadas conforme o avanço da dinâmica. As bibliotecas utilizadas podem ser consultadas através da Tabela 1.1.

Tabela 1 - Bibliotecas a serem importadas e utilizadas durante a execução do capítulo

BIBLIOTECA	DESCRIÇÃO DA FUNCIONALIDADE
PANDAS	Possibilita a leitura e manipulação de estruturas de dados através de arquivos contendo agrupamentos de dados.
MATPLOTLIB	Biblioteca de Dados 2d capaz de produzir imagens de alta qualidade através da análise dos dados fornecidos. Será utilizada para plotar gráficos estatísticos e os *word clouds* gerados na análise
NLTK	Plataforma voltada para trabalhar com dados gerados pela linguagem humana. Será utilizada para fazer o processamento de linguagem natural dos textos fornecidos.
SKLEARN	Biblioteca de aprendizado de máquina, contendo algoritmos de regressão, agrupamento e seleção para gerar diferentes tipos de aprendizado.

2.3.1. Descrição do Dataset

De maneira a categorizar o sentimento associado a um conjunto de palavras, é necessária alguma base de dados contendo diversos comentários e opiniões associados a algum tema. Uma das maneiras mais simples de conseguir montar estas bases é coletando dados de alguma página contendo avaliações de clientes, onde normalmente existe alguma nota associada a eles.

A base de dados a ser utilizada será a base do IMDb (*Internet Movie Database*), que possui *reviews* de filmes, músicas e programas de televisão em geral. Devido a grande massa de dados disponível em conjunto das notas associadas ao comentário do mesmo, a massa é muito conveniente e propícia para realizar uma mineração de opinião.

De modo a fazer a análise do texto utilizando o idioma em português, a comunidade de desenvolvedores na plataforma Kaggle traduziu o conteúdo dos comentários em inglês e adaptou para o português brasileiro. Para facilitar a manipulação de dados e o tamanho do arquivo a ser enviado durante a realização do capítulo, o *dataset* fornecido também foi adaptado, possuindo as seguintes colunas:

Tabela 2 - Colunas principais do dataset a ser fornecido para análise

COLUNA	DESCRIÇÃO
TEXT_PT	Texto do comentário traduzido para o português brasileiro de um programa ou filme assistido
SENTIMENTO	Representa a opinião geral, onde notas iguais ou maiores a 7 = “pos” e abaixo de 7 = “neg”
CLASSIFICACAO	Classificação do sentimento convertido para um valor binário, de modo que recebe 0 para sentimento positivo e 1 para sentimento negativo.

2.4. Execução dos Estágios

Após explicar os objetivos do curso e ajudar os alunos no preparo do ambiente de desenvolvimento, a parte prática será dividida em quatro etapas (estágios), detalhadas a seguir (Figura 2.1):

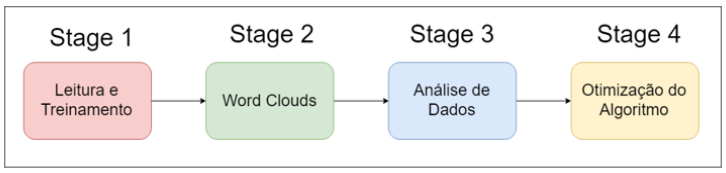

Figura 2.1 - Estrutura dos estágios a serem realizados no capítulo.

Cada etapa representará um dos passos realizados para a execução de uma análise de sentimento, de modo a facilitar o entendimento dos métodos de execução dividindo-os em partes. Além disso, um gabarito será disponibilizado contendo o conteúdo de cada parte ao término de sua explicação, de maneira a facilitar a identificação de possíveis erros durante a execução dos scripts por cada aluno. Cada etapa será detalhada sobre seus métodos e resultados a partir das próximas seções, sendo explicadas a seguir.

2.4.1. Stage 1

Nesse primeiro estágio, conforme a Figura 2.2**,** será iniciado o preparo do ambiente de desenvolvimento dos alunos através da leitura de uma base de dados previamente preparada para leitura e manipulação dos dados da mesma. Essa base de dados foi fornecida em um link para *download* no minicurso, possuindo uma base pré-processada e pronta para realizar a análise.

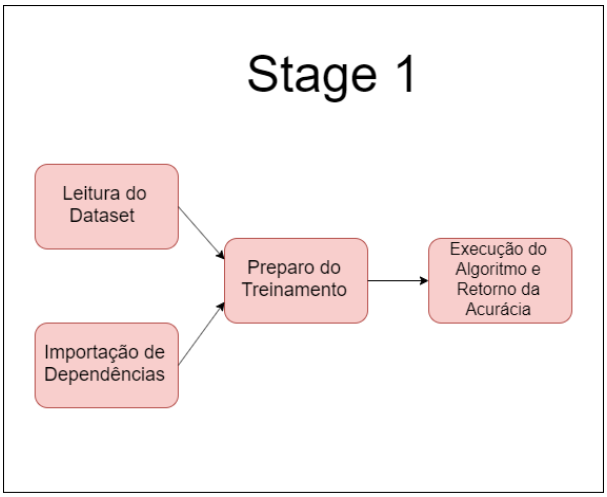

Figura 2.2 - Estrutura do primeiro estágio a ser realizado.

Após baixar a base de dados, será pedido aos participantes para criarem um novo notebook na plataforma *Google Colaboratory*, através de suas contas *Google* (Figura 2.3). Após criarem, os mesmos deverão fazer o upload da massa de dados baixada diretamente para o notebook criado, conforme a Figura 2.4, possibilitando a sua manipulação de dados.

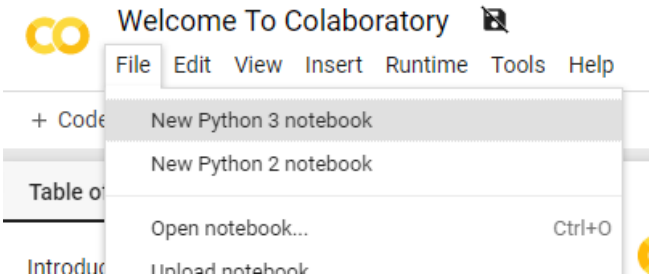

Figura 2.3 - Criação do arquivo a ser gerada a análise.

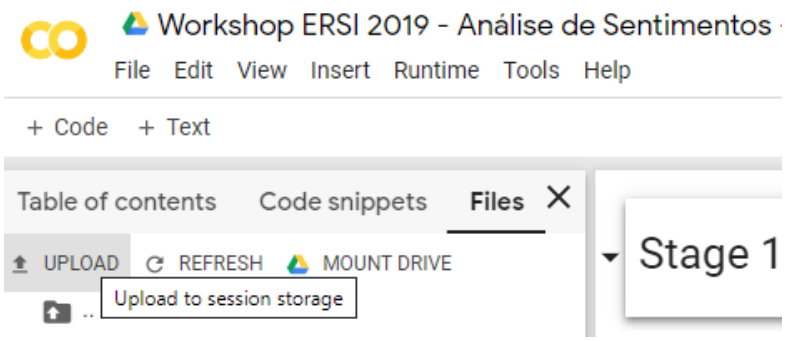

Figura 2.4 - Instruções de *upload* do *dataset* para possibilitar a manipulação de dados.

2.4.1.1. Utilizando a biblioteca Pandas para manipular dados

Com o ambiente e a massa preparados, é hora de começar a manipular os dados contidos na base. Realizando um comando para importar a biblioteca Pandas do *Python*, será possível ler os dados extraídos de um arquivo csv e convertê-lo para uma estrutura de dados propícia para habilitar a sua manipulação. Após converter esses dados em um *dataset*, é possível mostrar os resultados com a atribuição deste em uma variável, conforme mostrado na Figura 2.5.

	text_pt	sentiment	classificacao
0	Mais uma vez, o Sr. Costner arrumou um filme p...	neg	0
1	Este é um exemplo do motivo pelo qual a maiori...	neg	0
2	Primeiro de tudo eu odeio esses raps imbecis, ...	neg	0
3	Nem mesmo os Beatles puderam escrever músicas ...	neg	0
4	Filmes de fotos de latão não é uma palavra apr...	neg	0
5	Uma coisa engraçada aconteceu comigo enquanto ...	neg	0

Figura 2.5 - Primeira visualização dos dados do *dataset* na biblioteca Pandas.

2.4.1.2. *Bag-of-words*

Podendo manipular os dados, o próximo passo é converter todo o *corpus textual*, constituído por todos os comentários presentes no *dataset*, de maneira a criar uma representação para que o computador consiga interpretar estes dados. Uma das formas mais famosas para gerar essa visualização é através de uma abordagem onde todo o conteúdo do documento será representado como um vetor de palavras de acordo com suas ocorrências no mesmo [Matsubara et al., 2003]. Esse modelo de simplificação representativa é muito utilizado na área de Processamento de Linguagem Natural, sendo denominada *bag-of-words*.

A abordagem *bag-of-words* possibilita a representação de documentos textuais no formato de uma tabela atributo-valor, composta pelo número total de termos totais em cada uma de suas iterações. Dessa forma, todos os termos contidos no corpus textual tornam-se as colunas de um vetor, enquanto as linhas representam a frequência de cada uma das palavras contidas em cada iteração do mesmo. A presença de cada termo é categorizada de maneira binária, ou seja, se o termo está presente no documento, o valor

de uma posição Aij é acrescentado em 1, e caso contrário 0 [Matsubara et al., 2003] A Figura 2.6 ilustra a definição de um vetor utilizando a abordagem *bag-of-words*, de maneira a categorizar a frequência de todas as palavras presentes em cada frase do corpus textual de maneira binária.

	it	is	puppy	cat	pen	a	this
it is a puppy	1	1	1	0	0	1	0
it is a kitten	1	1	0	0	0	1	0
it is a cat	1	1	0	1	0	1	0
that is a dog and this is a pen	0	2	0	0	1	2	1
it is a matrix	1	1	0	0	0	1	0

Figura 2.6 - Exemplo de como os elementos são armazenados em uma *bag-of-words.*

2.4.1.3. Treinamento do algoritmo

Com o *array* montado, agora é possível analisar a frequência de todas as palavras em relação a todos os comentários. Dessa maneira, resta associar a frequência em conjunto ao sentimento associado, de maneira a verificar se as palavras associadas conseguem ser classificadas corretamente de acordo com o sentimento predominante na frase.

Para isso, será utilizado o conceito de aprendizado supervisionado previamente mencionado na seção 2.2.2, onde será separado 75% como massa de treino e 25% como massa de teste, de acordo com a função *train_test_split* da biblioteca SKLearn. Dessa forma, o *array* contendo a *bag-of-words*, em conjunto com a coluna de classificação do sentimento associado, irão compor as massas de treino e teste de acordo com esta proporção (Figura 2.7).

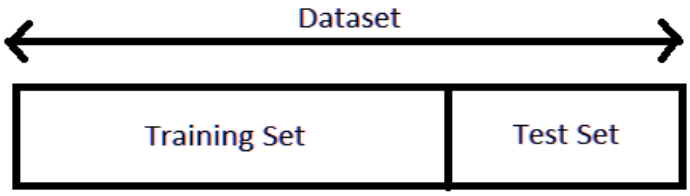

Figura 2.7 - Estrutura de uma base de treino e teste.

O último passo para finalizar este estágio é verificar a acurácia do modelo em si, de maneira a averiguar se os exemplos na massa de teste conseguem ser previstos corretamente baseados nos valores fornecidos pela massa de treino. Um dos métodos mais utilizados para conseguir prever os valores de uma resposta é o método da Regressão Linear, de modo que a resposta encontrada é baseada em um ou mais preditores [Hilbe, 2019]. Todavia, a utilização deste modelo proporciona uma resposta em uma variável contínua somente sendo baseada em valores numéricos, sendo assim incompatível com a análise dos termos no corpus textual por estarem em formato de texto.

Um outro modelo, denominado Regressão Logística, consegue realizar a previsão de variáveis discretas através de um valor binário fornecido, de maneira a

fornecer as chances de acerto dada determinada categoria no conjunto de dados. Devido à classificação binária do vetor de *bag-of-words*, é possível definir a acurácia do aprendizado supervisionado gerado utilizando este modelo, através da verificação dos valores na massa de teste com base na massa de treino (Figura 2.8).

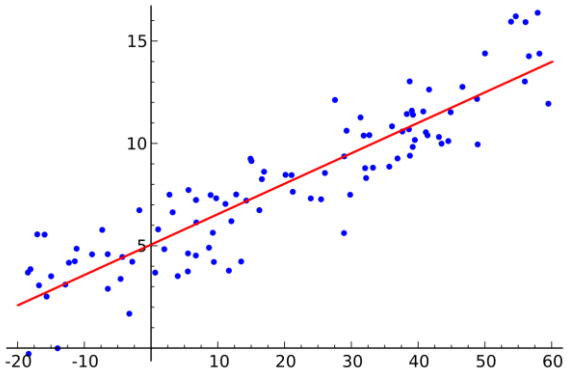

Figura 2.8 - Representação gráfica de uma Regressão Logística.

Desta forma, os valores binarizados no vetor *bag-of-words* são interpretados como variáveis discretas, sendo consequentemente preditores para a execução do algoritmo de Regressão Logística. Dessa forma, a porcentagem de acerto da regressão logística pode ser acessada através do método *score* da biblioteca SKLearn, baseada nos valores dados na coluna **"classificacao"** do *dataset* e nos valores separados na *bag-of-words*. A Figura 2.9 mostra a taxa de acerto obtida após a execução do algoritmo de Regressão Logística.

```
0.6664
```

Figura 2.9 - Primeira acurácia a ser obtida após a execução da Regressão Logística.

É importante destacar que neste capítulo está sendo aplicado o algoritmo de regressão já relatado devido sua simplicidade e objetivado para o objeto de estudo. No entanto, não é difícil encontrar trabalhos usando outros algoritmos como o Multinomial Naive Bayes (MNB) ou Support Vector Machine (SVM).

2.4.2. Stage 2

Nesse estágio, serão mostradas formas de representação dos dados obtidos através de imagens, de maneira a mostrar novas formas de interação e compreensão dos mesmos. A Figura 2.10 mostra um *overview* dos passos a serem realizados durante estes estágios.

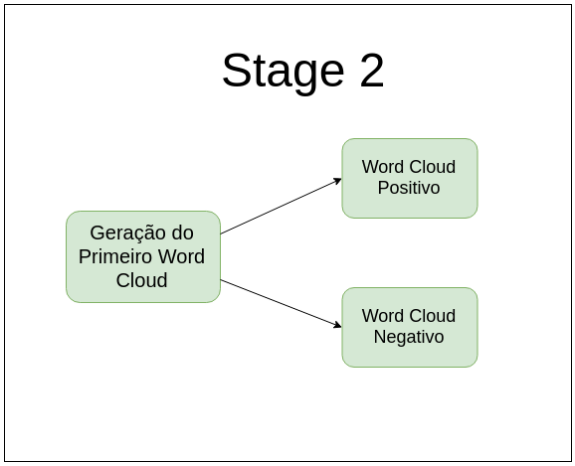

Figura 2.10 - Passos a serem realizados durante a execução do segundo estágio do *capítulo.*

2.4.2.1. *Word Cloud*

Ao gerar uma análise do corpus textual, é possível construir novas formas de visualização de dados para destacar a frequência das palavras geradas. Uma forma muito comum para desenvolver esse tipo de visualização é através de recursos gráficos, facilitando o entendimento dos resultados em geral. Segundo Lima (2008), o estudo da cultura visual em termos econômicos e tecnológicos pode proporcionar uma compreensão mais crítica em relação ao seu papel na contemporaneidade, facilitando o entendimento do tema e transcendendo o simples prazer visual que estas podem proporcionar.

Uma maneira para destacar a frequência de um termo é através de uma sumarização de texto, de forma a gerar um *overview* simples e intuitivo de um texto através do destaque de palavras que aparecem mais durante o mesmo [Heimerl et, al., 2014]. Isso normalmente é alcançado com o uso de recursos gráficos, como a mudança do tamanho da fonte de um texto que possui maior relevância ou possui um número maior de repetições presentes no mesmo.

Desta forma, pode-se ser introduzido o conceito de *word clouds*, que utiliza formas de percepção visual para facilitar a compreensão dos termos mais relevantes de maneira generalizada presentes em um corpus textual demasiadamente grande. Assim, um *word cloud* é representado por uma imagem contendo diversas palavras, onde a importância das mesmas é definida pelo tamanho de seu texto no canvas. A representação de um *word cloud* pode ser observada por meio da Figura 2.11.

Figura 2.11 - Um *word cloud* genérico (exemplo).

O objetivo principal nesse estágio é armazenar todas as palavras contidas na coluna de comentários e realizar uma contagem das mesmas, de modo a conseguir contabilizar o número de repetições feitas por uma delas. Ao contrário da representação no método *bag-of-words*, não será necessário armazenar as palavras de maneira binária, visto que o texto contido na palavra será essencial para representá-la no canvas da imagem a ser gerada.

Após a realização de um *loop* para armazenar todas as palavras em um vetor unidimensional, estes valores serão passados para a biblioteca *Python Wordclouds*, que será responsável por montar um objeto contendo uma figura com todas as palavras mais relevantes no mesmo. Essa imagem será plotada através da biblioteca Matplotlib, que irá definir as propriedades de tamanho e métodos de plotagem da figura. O primeiro resultado obtido pode ser visto na Figura 2.12:

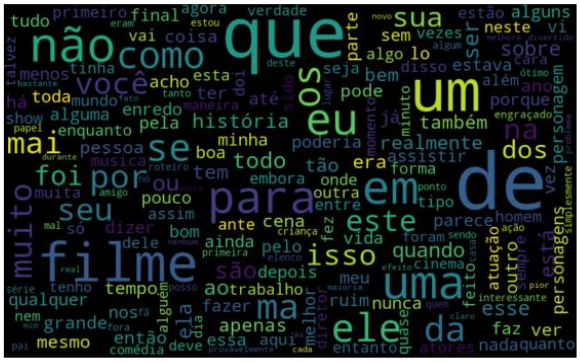

Figura 2.12 - Primeiro *word cloud* a ser gerado no segundo estágio.

Após gerar o primeiro *word cloud*, é proposto um desafio para aperfeiçoar as formas de visualização, transformando esses conhecimentos obtidos em uma função, e separando em dois tipos diferentes: um tipo para gerar *word clouds* somente para as palavras associadas a um sentimento positivo e outra para o sentimento negativo. O resultado pode ser observado nas Figuras 2.13 e 2.14.

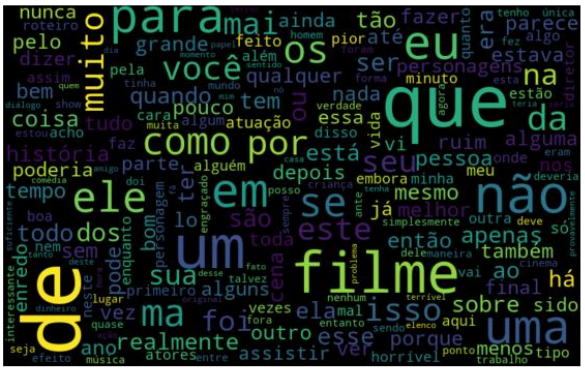

Figura 2.13 - *Word cloud* gerado contendo somente palavras associadas a um sentimento positivo.

Figura 2.14 - *Word cloud* gerado contendo somente palavras associadas a um sentimento *negative*.

2.4.3. Stage 3

Até agora, foram obtidas acurácia e uma forma de visualização em imagem, mas os resultados ainda não foram satisfatórios. A acurácia ainda possui uma precisão razoavelmente baixa, enquanto os *word clouds* mostraram valores semelhantes e contendo palavras em destaque como “de, que, um, os”.

Partindo desse ponto, é ideal que existam novas formas de visualização para analisar com maior precisão a frequência desses termos encontrados, de modo que venha a facilitar a geração de novas ideias para aperfeiçoar cada vez mais os algoritmos implementados. O objetivo desse estágio (Figura 2.15) é, então, proporcionar uma análise mais detalhada e estatística dos resultados obtidos.

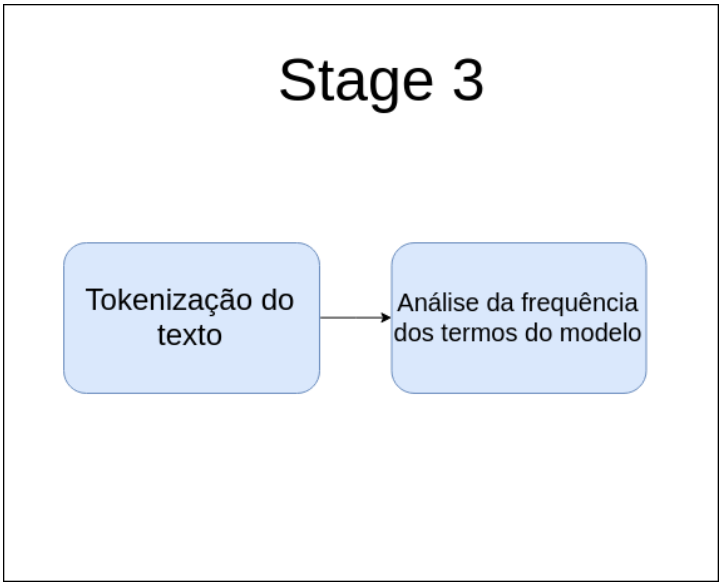

Figura 2.15 - Estrutura do terceiro estágio do capítulo.

2.4.3.1. Tokenização e análise estatística dos dados

Uma forma eficaz de ver a frequência dos termos é através da criação de outro *dataset*, contendo precisamente o número de repetições em cada comentário. Não será possível utilizar os valores armazenados de frequência armazenados dentro do objeto de *word cloud* gerado na seção 2.4.2.1, então será necessário utilizar outro método para gerar esse armazenamento.

Nesse caso, todas as palavras de uma frase precisam ser tokenizadas, ou seja, todo o corpus textual precisa ser convertido em lexemas. Um lexema é uma unidade de análise morfológica, sendo caracterizado como uma palavra, normalmente separada por um caractere de espaço dependendo do idioma [Chung e Gildea, 2009].

A partir desse método, todas as palavras podem ser caracterizadas, e assim possuírem seus valores agregados por suas frequências. Utilizando a biblioteca NLTK, esses valores podem ser somados e agregados no formato de um *array*, gerando uma nova coluna de frequência no novo *dataset*. Este resultado pode ser observado na Figura 2.16.

	Palavra	Frequencia
20	de	42441
14	que	32915
42	e	30045
3	o	24939
7	um	22420
102	a	21160
45	é	19511
200	em	13402
1	uma	13314
29	não	13003

Figura 2.16 - Primeira visualização da frequência dos dados obtidos em um novo *dataset.*

Esta visualização também pode ser feita através de um gráfico de barra. Reaproveitando o novo *dataset* obtido, é possível passar seus valores para serem plotados em uma figura através da biblioteca *Python Seaborn*, possibilitando uma visão estatística dos números obtidos de acordo com a quantidade desejada. Os resultados podem ser observados na Figura 2.17.

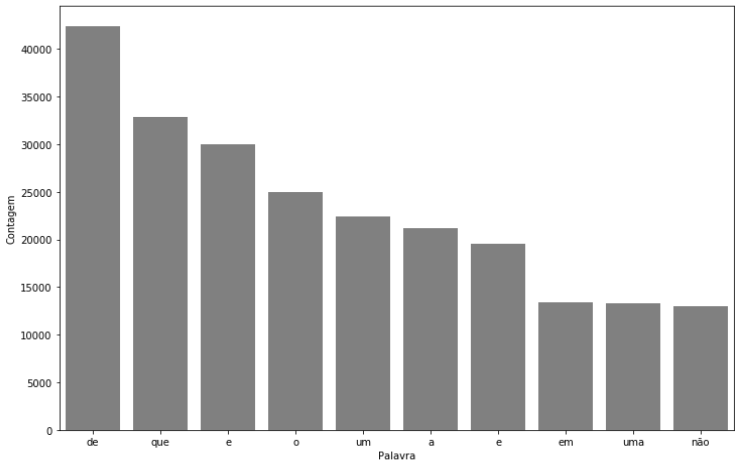

Figura 2.17 - Visualização da frequência dos dados em um gráfico de barra.

2.4.4. Stage 4

A etapa final consiste em otimizar a acurácia obtida na edição e remoção de trechos no texto do corpus textual que não agregam ao resultado desejado. Dessa forma, algumas das formas mais famosas para realizar o pré-processamento básico de um texto serão abordadas, conforme mostrado na Figura 2.18:

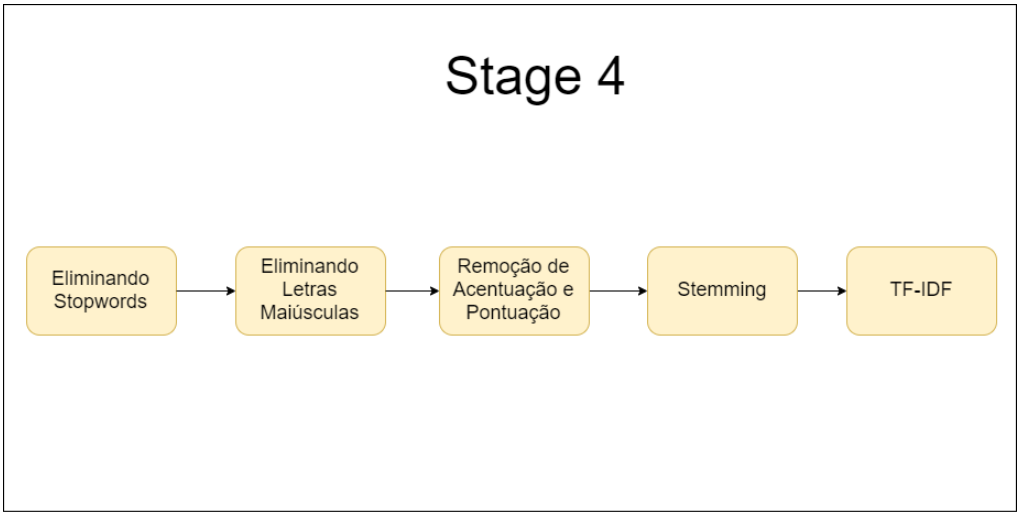

Figura 2.18 - Estrutura do quarto estágio do capítulo.

2.4.4.1. Remoção de *Stop words*

Após gerar os *word clouds* e os gráficos, foi possível ver que os termos de maior relevância associados a um sentimento são palavras que na língua portuguesa possuem uma frequência demasiadamente alta, como artigos, preposições e pronomes. Como esses termos não agregam ao objetivo a ser alcançado, não vale a pena que estes termos estejam contidos no modelo.

A solução é, então, realizar uma limpeza desses termos, sendo denominados *stop words*. Segundo E. Dragut et al. (2009), uma *stop word* é definida por palavras que não possuem um significado semântico relevante para um texto ou nas frases que aparecem. Dessa forma, a remoção das *stop words* do conjunto de palavras total irá facilitar e explicitar quais são os termos de principal relevância no modelo.

Utilizando a biblioteca *Stopwords* da biblioteca NLTK, é possível armazenar em uma variável todas as *stop words* associadas a determinado idioma, no caso desse curso, à língua portuguesa. Fazendo uma varredura das palavras e comparando se o termo presente é igual a uma *stop word*, é possível armazenar todos os termos não irrelevantes em um novo *array*, possibilitando a geração de outra coluna no *dataset* contendo somente os termos relevantes. Os resultados podem ser observados na Figura 2.19.

	text_pt	sentiment	classificacao	tratamento_1
0	Mais uma vez, o Sr. Costner arrumou um filme p...	neg	0	Mais vez, Sr. Costner arrumou filme tempo nece...
1	Este é um exemplo do motivo pelo qual a maiori...	neg	0	Este exemplo motivo maioria filmes ação mesmos...
2	Primeiro de tudo eu odeio esses raps imbecis, ...	neg	0	Primeiro tudo odeio raps imbecis, poderiam agi...
3	Nem mesmo os Beatles puderam escrever músicas ...	neg	0	Nem Beatles puderam escrever músicas todos gos...
4	Filmes de fotos de latão não é uma palavra apr...	neg	0	Filmes fotos latão palavra apropriada eles, ve...

Figura 2.19 - Nova coluna gerada no *dataset*, desta vez sem as *stopwords*.

2.4.4.2. Padronização do formato do texto

Ao gerar um gráfico de barra novamente (Figura 2.20), é possível ver que a palavra mais repetida na análise é “filme”, o que é intuitivo, devido a natureza desse *dataset*. Todavia, é possível ver que termos como “Eu, A, O” ainda estão sendo contabilizados e estão presentes na coluna nova, mesmo estas palavras sendo consideradas *stop words*.

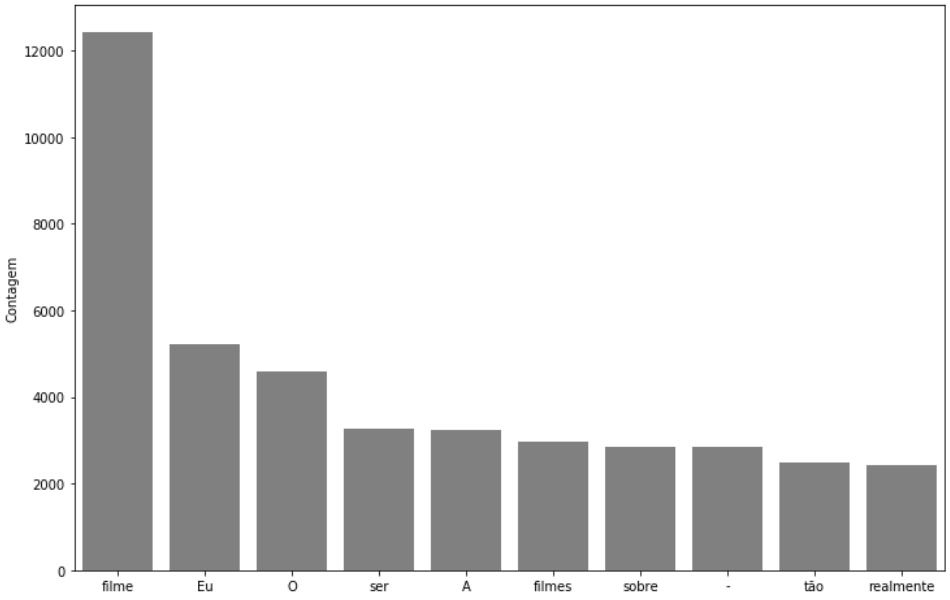

Figura 2.20 - Gráfico de barra contendo a nova frequência de palavras após a remoção dos *stop words.*

O que ocorre nesse caso é que o *array* contendo as *stop words* possui palavras em minúsculo, e a linguagem *Python* é uma linguagem *case sensitive*, que diferencia palavras com caracteres maiúsculos de minúsculos. Dessa forma, é necessário realizar outra varredura prévia antes de eliminar as *stop words,* convertendo todas as palavras em maiúsculo para minúsculo na coluna de comentário.

A segunda etapa de otimização visa então converter todo o texto, através da função nativa do *Python .lower().* Após essa transformação, a remoção das *stop words* pode ser feita novamente, dessa vez removendo todo o restante dos termos previamente em maiúsculo. Os resultados podem ser observados nas Figuras 2.21, 2.22 e 2.23.

tratamento_1	tratamento_2
Mais vez, Sr. Costner arrumou filme tempo nece...	vez, sr. costner arrumou filme tempo necessári...
Este exemplo motivo maioria filmes ação mesmos...	exemplo motivo maioria filmes ação mesmos. gen...
Primeiro tudo odeio raps imbecis, poderiam agi...	primeiro tudo odeio raps imbecis, poderiam agi...
Nem Beatles puderam escrever músicas todos gos...	beatles puderam escrever músicas todos gostass...
Filmes fotos latão palavra apropriada eles, ve...	filmes fotos latão palavra apropriada eles, ve...
Jma coisa engraçada aconteceu comigo enquanto ...	coisa engraçada aconteceu comigo enquanto assi...
Este filme terror alemão ser estranhos vi. Eu ...	filme terror alemão ser estranhos vi. ciente q...
Sendo fã longa data cinema japonês, esperava i...	sendo fã longa data cinema japonês, esperava i...
"Tokyo Eyes" fala menina japonesa 17 anos cai ...	"tokyo eyes" fala menina japonesa 17 anos cai ...
Fazendeiros ricos Buenos Aires têm longa polít...	fazendeiros ricos buenos aires têm longa polít...

Figura 2.21 - Nova coluna a ser inserida no *dataset*, após a transformação de todas as palavras para minúsculo.

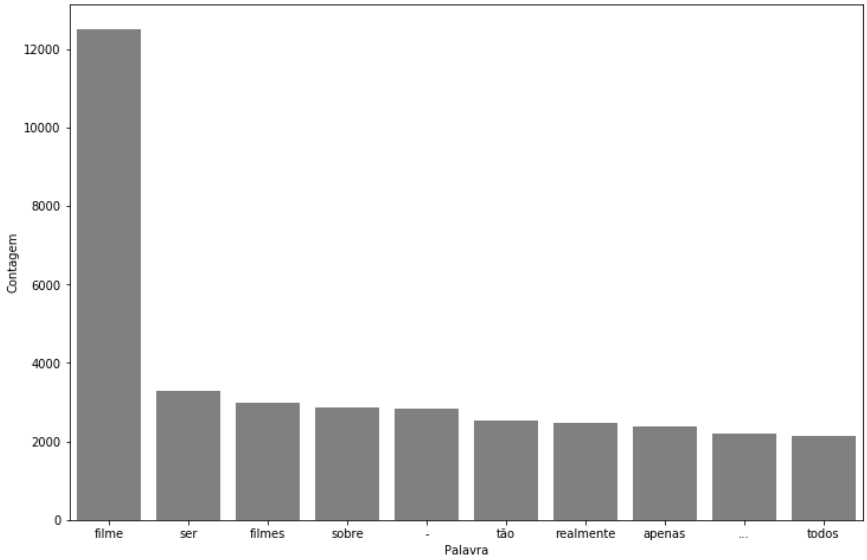

Figura 2.22 – Gráfico de barra gerado após a conversão das palavras no corpus textual para minúsculo.

Ao observar a figura 2.22, alguns problemas podem ser observados. Termos como "**-**" e "**...**" estão sendo sinalizados como os mais frequentes no corpus textual. Como estes termos não são palavras que agregam para o enriquecimento da análise, é conveniente remover todos os sinais de pontuação que permeiam o corpus textual, da mesma maneira que as *stop words* foram removidas.

Outra forma de otimização muito comum durante o pré-processamento do texto do corpus textual é a remoção da acentuação nas palavras fornecidas. Segundo Manning, et al, (2008), a remoção de acentos de palavras em idiomas como o inglês não realiza um grande impacto no significado da palavra, mantendo os seus significados originais e agrupando com sucesso todas as palavras no corpus textual. Todavia, em idiomas como o espanhol por exemplo, essa remoção pode comprometer significativamente no significado da palavra oferecida.

Apesar dos termos presentes na língua portuguesa possuírem palavras com significados diferentes devido à acentuação (como por exemplo as palavras "**secretaria**" e **secretária**", "**amém**" e "**amem**"), esse método é constantemente utilizado por conseguir agrupar palavras que possuem algum erro ortográfico referente a um erro de digitação ("**não**" por "**nao**", por exemplo), sendo assim consideradas válidas na frequência de somente um termo em si.

Dessa forma, ambas estas otimizações serão aplicadas no corpus textual, de maneira a conseguir remover os termos indesejados e agrupá-los com maior eficácia. Os resultados após a terceira otimização podem ser vistos na Figura 2.23.

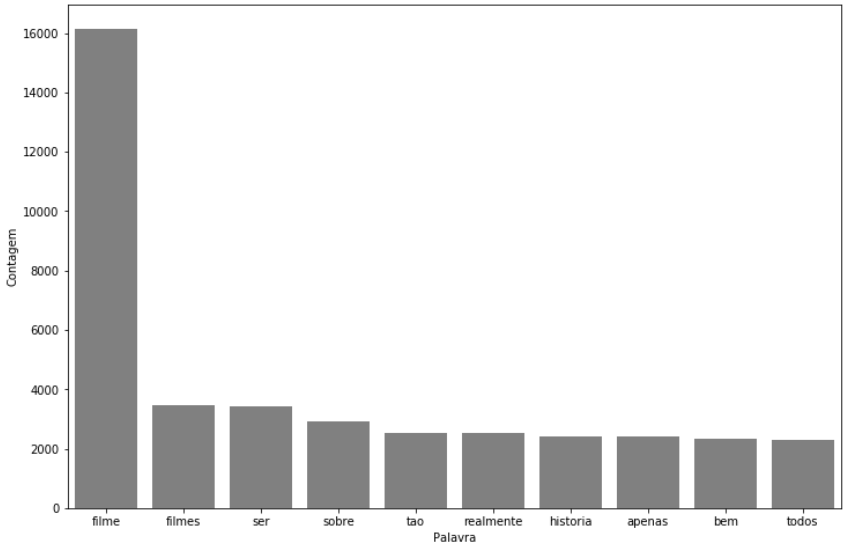

Figura 2.23 – Gráfico de barra gerado após a remoção de termos de pontuação e acentuação

tratamento_1	tratamento_2	tratamento_3
Mais vez, Sr. Costner arrumou filme tempo nece...	vez, sr. costner arrumou filme tempo necessári...	vez sr costner arrumou filme tempo necessário ...
Este exemplo motivo maioria filmes ação mesmos...	exemplo motivo maioria filmes ação mesmos. gen...	exemplo motivo maioria filmes ação mesmos géné...
Primeiro tudo odeio raps imbecis, poderiam agi...	primeiro tudo odeio raps imbecis, poderiam agi...	primeiro tudo odeio raps imbecis poderiam agir...
Nem Beatles puderam escrever músicas todos gos...	beatles puderam escrever músicas todos gostass...	beatles puderam escrever músicas todos gostass...
Filmes fotos latão palavra apropriada eles, ve...	filmes fotos latão palavra apropriada eles, ve...	filmes fotos latão palavra apropriada verdade ...
...	...	...
Como média votos baixa, fato funcionário locad...	média votos baixa, fato funcionário locadora a...	média votos baixa fato funcionário locadora ac...
O enredo algumas reviravoltas infelizes inacre...	enredo algumas reviravoltas infelizes inacredi...	enredo algumas reviravoltas infelizes inacredi...
Estou espantado forma filme maioria outros méd...	espantado forma filme maioria outros média 5 e...	espantado forma filme maioria outros média 5 e...
A Christmas Together realmente veio antes temp...	christmas together realmente veio antes tempo,...	christmas together realmente veio antes tempo ...
O drama romântico classe trabalhadora diretor ...	drama romântico classe trabalhadora diretor ma...	drama romântico classe trabalhadora diretor ma...

Figura 2.24 – Nova coluna no *dataset* gerada após a padronização de texto.

Figura 2.25 - *Word cloud* associado somente a palavras com sentimento positivo gerado após as otimizações

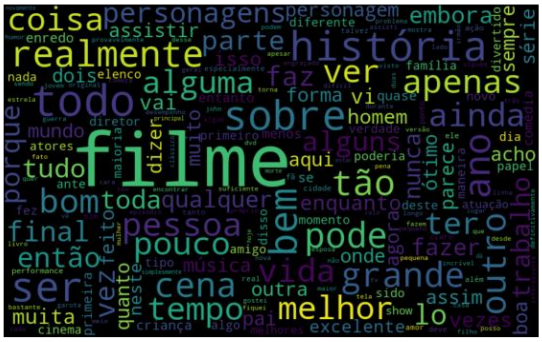

Figura 2.26 - *Word cloud* associado somente a palavras com sentimento negativo gerado após as otimizações.

2.4.4.3. Stemming

Ao verificar os resultados na Figura 2.23, é possível notar um caso peculiar. Após a padronização do corpus textual, a palavra “filme” e “filmes” são os dois termos com maior frequência. Todavia, ambas as palavras possuem o mesmo significado para a análise, de maneira que a segunda é somente uma derivação da primeira no plural. Dessa forma, pode-se dizer que o mesmo significado acaba ocupando duas posições no gráfico de termos mais relevantes, ocupando o espaço de termos com significado único que poderiam estar sendo representados no mesmo.

Ao analisar a estrutura morfológica de uma palavra, é possível ver casos em que a mesma é composta por uma estrutura similar, porém sendo derivada através de diferentes prefixos e sufixos. Estes muitas vezes não são de interesse imediato para realizar a contagem da frequência dos termos, visto que diferentes derivações resultam na contagem de termos distintos. Desta forma, seria conveniente à análise remover os sufixos de todas as palavras contidas no corpus, de maneira a aglutinar todas os termos com a mesma estrutura e significado.

O processo de remoção de todas as flexões das palavras de maneira a reduzi-las à mesma raiz de maneira computacional é denominado *Stemming*. Segundo Lovins (1968), a aplicação de *Stemming* ajuda a maximizar a utilidade dos

termos contabilizados, de maneira a otimizar as palavras agrupadas de acordo com um mesmo significado.

Dessa forma, a próxima etapa de otimização visa transformar todas as palavras do corpus textual e reduzi-las a uma mesma raíz (ou *stem*). Para isso, será utilizada um pacote da biblioteca NLTK denominada RSLPStemmer. O RSLP (Removedor de Sufixos na Língua Portuguesa) será responsável por converter todas as palavras no corpus textual em português brasileiro para seus respectivos radicais morfológicos, de maneira a otimizar a aglutinação dos termos.

Após realizar novamente a contagem dos termos no gráfico de barra (Figura 2.26), é possível notar uma mudança no total dos termos presentes. O novo termo "**film**" possui aproximadamente 20000 repetições, agrupando a frequência dos termos anteriores "**filme**" e "**filmes**". Outros termos, como '**tod**", subiram de posição, agrupando palavras com sufixos como "**todo**", "**todos**", "**todas**", etc. Dessa forma, é possível ter uma análise mais precisa de acordo com o significado real das palavras disponibilizadas no corpus.

tratamento_1	tratamento_2	tratamento_3	tratamento_4
Mais vez, Sr. Costner arrumou filme tempo nece...	vez, sr. costner arrumou filme tempo necessári...	vez sr costner arrumou filme tempo necessario ...	vez sr costn arrum film temp necessari alem te...
Este exemplo motivo maioria filmes ação mesmos...	exemplo motivo maioria filmes ação mesmos. gen...	exemplo motivo maioria filmes acao mesmos gene...	exempl motiv maior film aca mesm gener chat na...
Primeiro tudo odeio raps imbecis, poderiam agi...	primeiro tudo odeio raps imbecis, poderiam agi...	primeiro tudo odeio raps imbecis poderiam agir...	prim tud odei rap imbecil pod agir arm pressio...
Nem Beatles puderam escrever músicas todos gos...	beatles puderam escrever músicas todos gostass...	beatles puderam escrever musicas todos gostass...	beatl pud escrev music tod gost emb walt hill ...
Filmes fotos latão palavra apropriada eles, ve...	filmes fotos latão palavra apropriada eles, ve...	filmes fotos latao palavra apropriada verdade ...	film fot lata palavr apropri verdad tant ous q...
...	...	...	...
Como média votos baixa, fato funcionário locad...	média votos baixa, fato funcionário locadora a...	media votos baixa fato funcionario locadora ac...	med vot baix fat funcionari loc ach tud bem ",...
O enredo algumas reviravoltas infelizes inacre...	enredo algumas reviravoltas infelizes inacredi...	enredo algumas reviravoltas infelizes inacredi...	enred algum reviravolt infeliz inacredita enta...
Estou espantado forma filme maioria outros méd...	espantado forma filme maioria outros média 5 e...	espantado forma filme maioria outros media 5 e...	espant form film maior outr med 5 estrel men f...
A Christmas Together realmente veio antes temp...	christmas together realmente veio antes tempo,...	christmas together realmente veio antes tempo ...	christm togeth real vei ant temp cri john denv...
O drama romântico classe trabalhadora diretor ...	drama romântico classe trabalhadora diretor ma...	drama romantico classe trabalhadora diretor ma...	dram roman cl trabalh dire martin ritt tao ina...

Figura 2.27 – Nova coluna no *dataset* gerada após a remoção de sufixos.

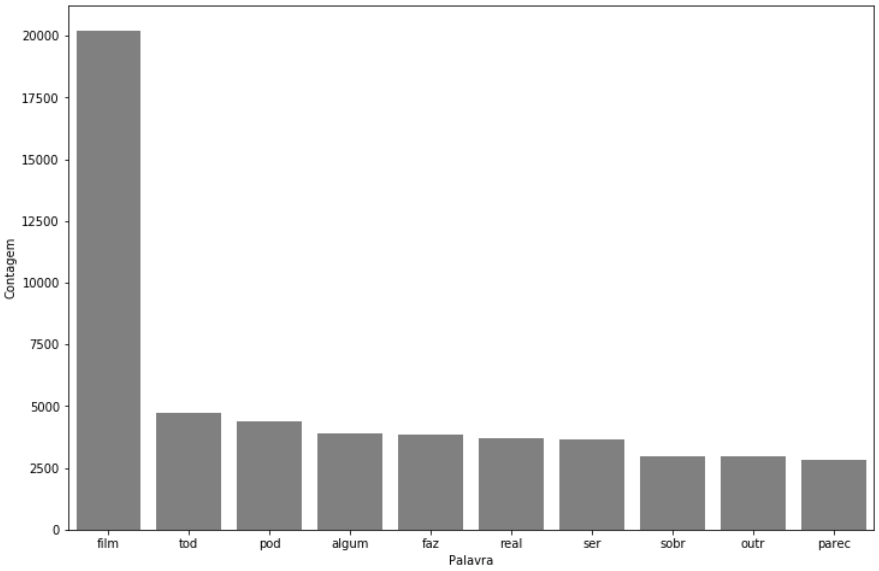

Figura 2.28 – Nova coluna no *dataset* gerada após a remoção de sufixos.

2.4.4.4. TF-IDF

Ao gerar a análise, é possível notar que palavras com maior frequência não são suficientes para demonstrar palavras de maneira polarizada, de modo que sejam diretamente associadas a algum sentimento. Analisando os dois *Word Clouds* gerados nas Figuras 2.25 e 2.26, nota-se que diversas palavras se repetem tanto associadas ao sentimento negativo quanto ao positivo. Dessa forma, é necessário gerar uma nova forma de análise, de modo a observar os termos que mais definem o sentimento na frase.

Uma forma de alcançar esse resultado é partindo do caminho oposto, ou seja, ao invés de analisar os termos mais frequentes, deve-se analisar os termos mais raros. Em frases como **"Eu achei o carro bonito"** e **"Eu achei o carro feio"**, as palavras **"bonito"** e **"feio"** são críticas para definir o sentimento da frase, apesar de estarem em uma frequência muito menor do que as outras palavras no corpus textual. Além disso, estas palavras têm muito mais chance de estarem presentes em frases associadas a somente um sentimento, aumentando assim a acurácia do modelo.

Dessa forma, a última etapa do capítulo será realizar a implementação de uma nova medida estatística que possibilite a análise de acordo com valores equilibrados pela frequência inversa dos termos, mais conhecida como TF-IDF. De acordo com Hiemstra (2000), o valor TF-IDF define que o peso dos termos presentes nos documentos deve ser proporcional à frequência dos termos em todo o corpus textual, e inversamente proporcional à sua frequência total em um documento. Dessa forma, é possível equilibrar a relevância de palavras comuns e evitar algumas palavras de serem mais importantes que outras.

Através da biblioteca SKLearn, é possível implementar essa nova ponderação, que irá equilibrar a frequência de todas as palavras no corpus textual de acordo com a quantidade de aparições. Dessa forma, palavras mais raras possuirão um peso maior,

enquanto palavras mais comuns receberão um peso menor, uniformizando o modelo de aprendizado.

Por fim, será gerado um aprendizado supervisionado da mesma forma feita no Stage 1, de modo a verificar se houve alguma mudança significativa na acurácia gerada. É possível também gerar um novo *dataset*, contendo o peso das palavras de acordo com a sua nova relevância de acordo com os seus valores TF-IDF (Figuras 2.29, 2.30).

	0
otim	3.180694
excel	2.400139
maravilh	1.743315
divert	1.621445
am	1.620834
incri	1.594425
gost	1.503948
favorit	1.487855
perfeit	1.452242
final	1.304025
vid	1.288748
muit	1.287829
mund	1.273158

Figura 2.29 - *Score* obtido de palavras positivas após uma iteração do algoritmo TF-IDF.

	0
ruim	-4.096869
pi	-3.652940
horri	-3.422551
terri	-2.879176
parec	-2.313847
nenhum	-2.205929
nad	-2.121887
chat	-1.995251
estup	-1.793116
minut	-1.743705
mal	-1.684672
dialog	-1.642053

Figura 2.30 - *Score* obtido de palavras positivas após uma iteração do algoritmo TF-IDF.

2.5. Discussão dos Resultados Obtidos

Na **Tabela 2.3** é possível ver todas as acurácias obtidas a partir das execuções dos algoritmos executados durante os estágios do capítulo.

Tabela 2.3 – Acurácias obtidas após os passos realizados durante o capítulo.

DESCRIÇÃO	ACURÁCIA
APÓS REGRESSÃO LOGÍSTICA COM CORPUS TEXTUAL INALTERADO (STAGE 1)	0.6664
APÓS REGRESSÃO LOGÍSTICA COM CORPUS TEXTUAL SEM *STOPWORDS* (STAGE 4)	0.6808
APÓS REGRESSÃO LOGÍSTICA COM CORPUS TEXTUAL SEM *STOPWORDS* E EM LOWERCASE (STAGE 4)	0.664
APÓS REGRESSÃO LOGÍSTICA COM CORPUS TEXTUAL SEM *STOPWORDS*, EM LOWERCASE E COM PONTUAÇÃO E ACENTUAÇÃO REMOVIDA (STAGE 4)	0.6792
APÓS REGRESSÃO LOGÍSTICA COM CORPUS TEXTUAL SEM *STOPWORDS*, EM LOWERCASE, COM PONTUAÇÃO E ACENTUAÇÃO REMOVIDAS E STEMMING APLICADO (STAGE 4)	0.6936
APÓS REGRESSÃO LOGÍSTICA PADRONIZADA COM VALORES TF-IDF, COM CORPUS TEXTUAL SEM *STOPWORDS*, EM LOWERCASE, COM PONTUAÇÃO E ACENTUAÇÃO REMOVIDAS E STEMMING APLICADO (STAGE 4)	0.884

Após realizar algumas otimizações, é possível ver que em alguns casos a acurácia acabou diminuindo. Esses casos são comuns, visto que ao aproximar o corpus textual de padronização mais homogênea muitas vezes pode acarretar numa maior taxa de erro, dependendo do tamanho e da forma do cálculo da acurácia do seu *dataset*.

É importante ressaltar o aumento significativo do aumento da acurácia após a implementação dos valores TF-IDF. A análise dos termos mais raros é muitas vezes uma das formas mais eficazes para obter informações pertinentes referentes ao seu conjunto de informações. Ainda assim, é importante verificar sob diferentes perspectivas, de maneira a expandir o número de conclusões a serem tiradas a partir do seu conjunto de dados.

A aplicação de *Stemming* nas palavras também é um fator importante a ser discutido. Embora o agrupamento dos termos derivados tenha contribuído para o aumento da acurácia em si, a representação das novas palavras definidas somente por sua raiz muitas vezes gera uma considerável perda semântica, tornando o entendimento

das mesmas mais difícil. Dessa forma, a visualização desse novo corpus textual acaba não sendo totalmente intuitiva para a representação gráfica em um *Word cloud*.

Por fim, é importante ressaltar que os resultados são afetados diretamente pelo tamanho de registros no *dataset* e da *seed* fornecida para separar os dados. No caso do capítulo, o número de dados foi consideravelmente menor, de maneira a facilitar o processamento de dados realizando-os de uma maneira mais ágil. De acordo com o número de informações do *dataset*, é possível obter uma classificação mais abrangente, consequentemente aumentando a precisão a ser obtida.

2.6. Conclusão

Este capítulo apresentou e discutiu os conceitos base na área de ciência de dados, de modo a possibilitar um primeiro contato no tema de Análise de Sentimento. Desta forma, foi necessário implementar alguns algoritmos e métodos léxicos de maneira a assegurar uma acurácia significativa para o modelo proposto.

É importante ressaltar que o *dataset* utilizado foi previamente montado e adaptado para facilitar a manipulação dos dados, e assegurar o foco somente no ensino dos conceitos base. Outras bases de dados possuem um número de registros muito maior, necessitando de mais tempo de processamento para a execução dos comandos. Além disso, estes necessitam de um tratamento prévio para possibilitar o uso de algoritmos como o de regressão logística e afins.

Além disso, existem outras formas de conseguir alcançar o mesmo resultado obtido, porém através de abordagens diferentes. Neste capítulo, foi abordado a técnica de Aprendizado Supervisionado, porém é possível obter resultados similares utilizando Aprendizados Semi ou Não Supervisionados. Dessa forma, é essencial aprender sobre as necessidades e recursos que seu projeto irá possuir, de modo a implementar o método que melhor se encaixe para o sucesso do mesmo.

Referências

Agarwal, B., Mittal, N., Bansal, P., & Garg, S. (2015). *Sentiment Analysis Using Common-Sense and Context Information. Computational Intelligence and Neuroscience, 2015, 1–9.*

Benevenuto, F., Ribeiro, F. and Araújo, M. (2015) *Métodos para Análise de Sentimentos em Mídias Sociais.*, Short course in the Brazilian Symposium on Multimedia and the Web (Webmedia).

Bing, Liu (2010) Sentiment Analysis and Subjectivity , 2nd edn., Handbook of Natural Language Processing.

Ceci, F., Alvarez, G., Gonçalves, A. (2017) *Análise de sentimento e mineração de opinião: Uma revisão bibliométrica da literatura*, Revista Espacios, Vol. 38 (Nº 14).

Chung, T. and Gildea, D. (2009) *Unsupervised tokenization for machine translation*, EMNLP 2009 - Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: A Meeting of SIGDAT, a Special Interest Group of ACL, Held in Conjunction with ACL-IJCNLP 2009: .

Dragut, E., Fang, F., Sistla, P., Yu, C. and Meng, W. (2009) *Stop word and related problems in web interface integration*, Proceedings of the VLDB Endowment.

Dumais, S., Platt, J., Hecherman, D. & Sahami, M. (1998) *Inductive Learning Algorithms and Representations for Text Categorization*, Proceedings of the seventh international conference on Information and knowledge management: .

Franz, M., Hillman, J. (2016) *A Tipologia de Jung - Ensaios sobre Psicologia Analítica*, 2nd edn., Cultrix.

Heimerl, F., Lohmann, S., Lange S. and Ertl, T. (2014) *Word cloud explorer: Text analytics based on word clouds*, Institute for Visualization and Interactive Systems (VIS)

Hiemstra, D. (2000). *A probabilistic justification for using tf×idf term weighting in information retrieval. International Journal on Digital Libraries, 3(2), 131–139.*

Hilbe, J. M. (2009). *Logistic regression models*. Chapman and hall/CRC.

Horta, E. G. (2015) *Aplicação de Máquinas de Aprendizado Extremo ao Problema de Aprendizado Ativo.* Programa de PósGraduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, UFMG.

Liddy, E.D. (2001) *Natural Language Processing*, 2nd edn., Encyclopedia of Library and Information Science: Marcel Decker, Inc.

Lima, C. (2008) *O uso da leitura de imagens como instrumento para a alfabetização visual.* Cadernos PDE, Vol. II. Curitiba.

Lovins, J. (1968) *Developing of a Stemming Algorithm* - Mechanical Translation and Computational Linguistics

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Matsubara, E., Martins, C., Monard, M. (2003) *PreTexT: uma ferramenta para pre-processamento de textos utilizando a abordagem bag-of-words*., Relatórios Técnicos do ICMC - São Carlos.

Mohammad, S. and Turney, P. (2013) *Crowdsourcing a Word–Emotion Association Lexicon*, Institute for Information Technology, National Research Council Canada: .

Monard, M. and Baranauskas, J. (2003) *Conceitos sobre aprendizado de máquina*, 1.1 edn., Sistemas inteligentes-Fundamentos e aplicações.

Reis, J.; Benevenuto, F.; de Melo, P. V.; Prates, R.; Kwak,H.; and An, J. (2015). *Breaking the news: First impressionsmatter on online news*. In Proceedings of the ICWSM.

Reis, J.; Gonçalves, P.; Vaz de Melo, P. O.; Prates, R.; and Benevenuto, F. 2014. *Magnet news: You choose the polarity of what you read*. Proceedings of ICWSM

Sebastiani, F (2002) *Machine learning in automated text categorization*, ACM Computing Surveys.

Serrano-Guerrero, J., Olivas, J., Romero, F. Herrera-Viedma, E. (2015) *Sentiment analysis: A review and comparative analysis of web services*, 2nd edn., Information Sciences, v. 311, p. 1838.

Zhang, K., Cheng, Y., Liao, W., & Choudhary, A. (2012). *Mining millions of reviews. Proceedings of the 13th International Conference on Electronic Commerce - ICEC '11*.

Biografia Resumida dos Autores

André Viana Tardelli

André é graduando do curso de Ciência de Computação na Universidade Federal do Rio de Janeiro. Atualmente atua como instrutor e desenvolvedor no Grupo Caelum, ministrando cursos presenciais nos temas de Front End e Data Science. Seu foco de pesquisa, atualmente contendo artigos apresentados em conferências nacionais e internacionais, envolve implementar conceitos de psicologia aplicados à tecnologia, buscando novas formas de humanizar as interações realizadas de maneira digital.

Lattes: http://lattes.cnpq.br/8627393261758849

Angélica Fonseca da Silva Dias

É doutora em Informática pela Universidade Federal do Rio de Janeiro (PPGI - 2018) com ênfase em Gestão de Sistemas Complexos. Mestre em Sistemas de Informação pela UFRJ. MBA em Gestão Executiva e E-Business pela COPPEAD e Inteligência e Database Marketing, Aperfeiçoamento em Gerência Avançada de Projetos/NCE/UFRJ. Graduação em Processamento de Dados/UNESA. Foi Diretora da Área de Extensão do Instituto Tércio Pacitti/UFRJ, Coordenadora Acadêmica dos Cursos de pós-graduação na UFRJ e atua como Professor Convidado no programa de pós-graduação em informática, Instituto de Economia e HCTE da UFRJ com a orientação e coorientação de alunos de graduação, pós-graduação e mestrado. Tem experiência nas áreas de Administração Pública, Gerência de Projetos, Ciência da Computação e Educação. Temas de interesse: Gestão de Conhecimento, Social Computing, Economia Circular Computacional, Data Science, Data Literacy, Trabalho e Aprendizagem Cooperativa apoiada por computador (CSCW e CSCL), Tecnologia Assistiva, Gestão Estratégica da Informação e Educação a distância. Lattes: http://lattes.cnpq.br/8795875378897586

Juliana Baptista dos Santos França

É doutora em Informática pela Universidade Federal do Rio de Janeiro (PPGI/UFRJ - 2018) com ênfase em Gestão de Sistemas Complexos. Finalizou seu Pós doutorado na UFRJ na área de CSCW (PPGI/UFRJ - 2018). Mestre em Informática pelo Programa de Pós-Graduação em Informática (PPGI/UNIRIO - 2012) da Universidade Federal do Estado do Rio de Janeiro, com ênfase na linha de pesquisa Sistemas de Apoio a Negócios. Possui graduação em Sistemas de Informação pela Universidade Federal do Estado do Rio de Janeiro (UNIRIO), e também graduação em Engenharia Cartográfica pela Universidade do Estado do Rio de Janeiro (UERJ). Atualmente é Professora Adjunto na Universidade Federal Rural do Rio de Janeiro (UFRRJ) junto ao Departamento de Computação (DECOMP) em Banco de Dados e atua como colaboradora no programa de pós-graduação em informática da UFRJ com a coorientação de alunos de mestrado. Atuou na organização de eventos científicos nacionais e internacionais como ISCRAM 2016, Summer School em IHC/CSCW 2019, e SBSC 2019. Tem experiência na área de Sistemas de Informação e atua principalmente nos seguintes temas: Colaboração (CSCW), Gestão de Conhecimento, Processos de Negócio, Aprendizagem Colaborativa, Suporte à Decisão e Modelagem Conceitual e Ontológica. Lattes: http://lattes.cnpq.br/9341068095520817