UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Métodos de Quantificação *Classify and Count* (CC) para Análise de Sentimentos

Guilherme Brunassi Nogima

São Carlos – SP

Métodos de Quantificação *Classify and Count* (CC) para Análise de Sentimentos

Guilherme Brunassi Nogima

Orientador: **Prof. Dr. Ricardo Marcondes Marcacini**

Monografia final de conclusão de curso apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.
Área de Concentração: Inteligência Artificial

USP – São Carlos
Junho de 2021

Nogima, Guilherme Brunassi
Métodos de Quantificação *Classify and Count* (CC) para Análise de Sentimentos / Guilherme Brunassi Nogima. - São Carlos - SP, 2021.
40 p.; 29,7 cm.

Orientador: Ricardo Marcondes Marcacini.
Monografia (Graduação) - Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP, 2021.

1. Análise de sentimento. 2. Quantificação. 3. Classificação. I. Marcacini, Ricardo Marcondes. II. Instituto de Ciências Matemáticas e de Computação (ICMC/USP). III. Título.

AGRADECIMENTOS

Em primeiro lugar, agradeço imensamente ao meu orientador Ricardo Marcacini, que ao longo do semestre prestou todo auxílio necessário, sendo muito atencioso e paciente sempre que precisei de sua ajuda. Também devo muitos agradecimentos a meu pai, minha mãe e irmã, que sempre me apoiaram incondicionalmente nas minhas decisões e sempre confiaram muito em mim. Por fim, um agradecimento especial a todos os amigos que fiz durante minha trajetória na universidade, com quem vivi momentos inesquecíveis e que vou levar para toda vida.

RESUMO

NOGIMA, G. N.. **Métodos de Quantificação *Classify and Count* (CC) para Análise de Sentimentos**. 2021. 40 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

A quantificação é um procedimento que busca determinar como estão distribuídas as classes ou categorias em um conjunto de dados. Existem inúmeras aplicações reais, uma delas é na área de análise de sentimento, na qual, a partir de dados textuais e um classificador, é possível estimar o sentimento geral em relação a um tópico e determinar se há uma prevalência positiva ou negativa. Um dos métodos de quantificação mais conhecidos é o *Classify and Count* (CC), que realiza uma operação trivial de classificar os dados e contá-los. Diversos métodos mais sofisticados foram desenvolvidos, levando o CC a condição de um método pouco eficiente. Por outro lado, estudos recentes apresentaram novos resultados a favor de métodos CC de quantificação. Nesses estudos é demonstrado que um protocolo adequado para seleção de modelos e otimização de parâmetros orientado a métricas de quantificação, consegue tornar o CC competitivo em relação a outros métodos mais robustos. Nesse sentido, o objetivo deste trabalho é avaliar métodos CC de análise de sentimentos considerando um protocolo de otimização para quantificação. Em especial, foi apresentado um novo quantificador baseado no BERT, um modelo de representação de linguagem introduzido no ano de 2018 pelo time de pesquisadores do Google. Uma das motivações da proposta é avaliar uma estratégia que obtenha bons resultados tanto na classificação quanto na quantificação, permitindo reuso de modelos. Os resultados experimentais realizados em três conjuntos de *benchmark* mostraram que o modelo desenvolvido alcançou resultados relevantes, sendo que, em muitos casos, obteve, tanto em testes de quantificação, quanto em testes de classificação, a melhor performance quando comparado a outros classificadores.

Palavras-chave: Análise de sentimento, Quantificação, Classificação.

ABSTRACT

NOGIMA, G. N.. **Métodos de Quantificação *Classify and Count* (CC) para Análise de Sentimentos**. 2021. 40 f. Monografia (Graduação) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Quantification is a procedure that aims to determine the distribuition of classes within a dataset. There are many real life applications, one of them is in the field of sentiment analysis, in which, from textual data and a classifier, it is possible to estimate the general feeling regarding a topic and determine whether there is a positive or negative prevalence. One of the best known quantification methods is the *Classify and Count* (CC). Several more sophisticated methods were developed, bringing CC to a condition of an inefficient method. On the other hand, recent studies have presented new results in favor of CC quantification methods. These studies show that an adequate protocol for model selection and parameter optimization oriented to quantification metrics can make CC competitive in relation to other more robust methods. In this sense, the objective of this work is to evaluate CC methods of sentiment analysis considering a quantification optimization protocol. In particular, a new quantifier based on BERT, a language representation model that has obtained state-of-the-art results in sentiment analysis, is presented. One of the motivations of the proposal is that it evaluates a strategy that obtains good results in both classification and quantification, allowing the reuse of models. The experimental results performed on three datasets showed that the developed model achieved relevant results, and in many cases it obtained, both in quantification and classification tests, the best performance when compared to others classifiers.

Key-words: Sentiment Analysis, Quantification, Classification.

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACC ..... Adjusted Classify and Count
AE ..... Absolute Error
BERT .... Bidirectional Encoder Representations from Transformers
CC ..... Classify and Count
PACC .... Probabilistic Adjusted Classify and Count
PCC ..... Probabilistic Classify and Count
PLN ..... Processamento de Linguagem Natural
RAE ..... Relative Absolute Error

LISTA DE SÍMBOLOS

$\oplus$ — classe positiva

$\ominus$ — classe negativa

h — Modelo de classificação *hard*

s — Modelo de classificação *soft*

p — Prevalência real da classe positiva

$\hat{p}$ — Prevalência estimada da classe positiva

n — Prevalência real da classe negativa

$\hat{n}$ — Prevalência estimada da classe negativa

U — Conjunto de dados não categorizados

SUMÁRIO

Capítulo 1

INTRODUÇÃO

1.1 Motivação e Contextualização

A quantificação possui por objetivo retornar as estimativas da frequência relativa ou a prevalência das classes de interesse em um conjunto de dados não categorizados. Isso é atingido por meio do treinamento de um estimador a partir de dados categorizados [12]. Dessa forma, a quantificação obtém uma estimativa agregada para um conjunto de teste, de modo que não há a necessidade de se obter predições individuais para cada unidade do conjunto [7].

Por outro lado, o objetivo da classificação é retornar uma previsão correta para cada instância de um conjunto de dados não categorizados. Sendo assim, a principal diferença entre a quantificação e a classificação é o objeto de foco de cada uma, enquanto a quantificação busca fazer previsões sobre amostras, ou um conjunto de instâncias, a classificação busca fazer previsões para instâncias individuais [6].

Um dos métodos de quantificação mais difundidos atualmente é o *Classify and Count (CC)*, uma estratégia simples e intuitiva que classifica cada uma das instâncias e conta os resultados atribuídos a cada uma das classes [12]. Outros métodos populares são algumas das variantes do CC:

- *Adjusted Classify and Count (ACC)*, um método que realiza correções no CC com base nos indicadores *True Positive Rate* (*TPR*) e *False Positive Rate* (*FPR*) [7], assumindo um problema binário;
- *Probabilistic Classify and Count (PCC)*, que utiliza um classificador que retorna a probabilidade de uma instância pertencer a uma dada classe;
- *Probabilistic Adjusted Classify and Count (PACC)*, que agrega os dois métodos citados anteriormente [6].

O método CC, quando comparado a outros métodos de quantificação, tende a apresentar uma performance ruim, e isso tende a se acentuar em situações em que a distribuição de classes possui uma diferença significativa entre as fases de treinamento e teste [7]. A razão disso é que grande parte dos métodos de aprendizado supervisionado assume que a distribuição das categorias é independente e identicamente distribuída (*IID*). Dessa forma, como o objetivo da

classificação é otimizar medidas de erro de classificação, o CC não é considerado um bom estimador quando o objetivo é otimizar medidas de quantificação [10].

Um estudo recente apresentado por Moreo e Sebastiani (2021) [12], intitulado "*Re-assessing the "Classify and Count" Quantification Method*", apresentou novos resultados a favor de métodos CC de quantificação. Os autores discutiram que grande parte dos trabalhos que realizam experiências de quantificação com o método CC não conduzem uma otimização que busque adequá-lo à realidade da quantificação, com base em medidas pertinentes para quantificação. Os resultados de experiências que realizam essas otimizações mostram que métodos de quantificação baseados em CC podem ser competitivos em relação a outros métodos mais robustos, tais como o ACC, PCC e PACC [12]. Esse resultado mostra que é possível o reuso de modelos de classificação para problemas de quantificação, reduzindo o esforço para manutenção de modelos em cenários práticos.

A aplicação de interesse neste trabalho é a análise de sentimentos em dados textuais utilizando quantificação binária, de modo a atribuir uma característica positiva ou negativa ao conteúdo do texto. Nesse sentido, algumas das aplicações reais deste tipo de problema são estimar a satisfação geral do público em relação a um produto a partir de avaliações ou determinar a popularidade de um candidato em eleições, a partir de *tweets* [9].

Determinar se um texto possui uma opinião positiva ou negativa é uma tarefa bastante subjetiva, uma vez que uma mesma opinião pode transmitir diferentes sentimentos em relação a um assunto, condensando argumentos favoráveis e desfavoráveis em um mesmo trecho [18]. Nesse contexto, existem diversos classificadores que tentam automatizar essa tarefa, de modo a permitir a análise textual em larga escala. Um desses classificadores, que será discutido ao longo deste trabalho, é baseado no *BERT (Bidirectional Encoder Representations from Transformers)* [3].

Introduzido no ano de 2018 pelo time de pesquisadores do Google, o BERT é um modelo de representação de linguagem que difere de seus pares mais recentes ao aplicar um treinamento bidirecional, considerando, dessa forma, o contexto de uma palavra dos lados direito e esquerdo simultaneamente. Também vale destacar como diferencial a estratégia de treinamento do BERT, baseado na previsão de palavras mascaradas e na predição de próximas sentenças. O BERT usa uma arquitetura denominada *Transformers*, que permite seu treinamento em grandes bases de textos a um custo computacional menor do que abordagens anteriores. O BERT foi capaz de melhorar os resultados em diversas tarefas de Processamento de Linguagem Natural (NLP), como a análise de sentimento e a resposta de perguntas [8], obtendo desempenho próximo de humanos.

Uma vez que classificadores baseados no BERT obtêm resultados estado-da-arte para tarefas de análise de sentimentos [5, 13] e que o problema de quantificação CC tem alta dependência do modelo de classificação, a seguinte questão de pesquisa é levantada: quantificação do tipo CC para análise de sentimentos usando modelos BERT obtém melhores resultados do que

métodos existentes?

1.2 Objetivos

Tendo em vista as divergências apresentadas relacionadas à aplicabilidade do método CC para tarefas de quantificação em análise textual, este trabalho visa a propor uma implementação do CC, utilizando o BERT como classificador e otimizando os parâmetros do BERT considerando métricas de quantificação. Sendo assim, os objetivos do trabalho apresentam-se, essencialmente, de duas formas, a primeira delas consiste na utilização de um classificador adequado para o problema, comparando o BERT com os demais classificadores utilizados em experimentos anteriores. E em uma segunda parte, é utilizado um protocolo de otimização adequado para uma tarefa de quantificação.

Além disso, um fator importante considerado neste trabalho consiste no desenvolvimento de um modelo que tenha um bom desempenho não apenas nas tarefas de quantificação, mas que seja também um bom classificador para análise de sentimento. Dessa forma, o modelo, treinado uma única vez, pode entregar resultados satisfatório de duas formas e ser reutilizado em diferentes etapas de um *pipeline* de análise de sentimentos.

Os experimentos realizados neste trabalho baseiam-se no trabalho de Moreo e Sebastiani (2021) [12], trazendo incorporações que permitam a utilização do BERT, juntamente com uma otimização especificamente desenvolvida para esse classificador.

1.3 Organização

O Capítulo 2 irá apresentar os conceitos envolvidos no desenvolvimento do trabalho. O Capítulo 3 contempla a metodologia aplicada, abordando o passo a passo para obtenção dos resultados e detalhando como foi realizada a implementação. No Capítulo 4, os resultados são apresentados e discutidos. Por fim, o Capítulo 5 traz uma conclusão sobre o trabalho, bem como considerações gerais sobre o curso de Engenharia de Computação do ICMC/USP.

Capítulo 2

MÉTODOS, TÉCNICAS E TECNOLOGIAS UTILIZADAS

2.1 Considerações Iniciais

Neste capítulo, são abordados os conceitos e métodos envolvidos no desenvolvimento do trabalho. A notação utilizada é apresentada a seguir. h e s representam modelos de classificação *hard* e *soft*; p e $\hat{p}$ denotam a prevalência real e estimada da classe positiva; n e $\hat{n}$ denotam a prevalência real e estimada da classe negativa; $\oplus$ e $\ominus$ representam as classes positivo e negativo e U denota um conjunto de dados não categorizado.

2.2 Conceitos e Técnicas Relevantes

2.2.1 Quantificação

Um problema típico de classificação busca atribuir uma categoria para cada um dos exemplos em um conjunto de teste, portanto, um modelo de classificação pode ser definido da seguinte forma:

$$h : X \rightarrow \{c_1, ..., c_l\} \tag{2.1}$$

na qual X corresponde ao espaço de entrada que representa individualmente os exemplos a serem classificados, e o conjunto $\{c_1, ..., c_l\}$ representa as possíveis categorias que os exemplos podem assumir [1].

Em contrapartida, um problema de quantificação busca desenvolver um modelo capaz de estimar as prevalências de cada classe na amostra, que pode ser definido como:

$$h : X^n \rightarrow [0, 1]^l \tag{2.2}$$

Sendo na qual cada elemento do vetor de estimativas representa a probabilidade da classe j na

amostra, ou seja,

$$\sum_{j=1}^{l} \hat{p}_j = 1 \tag{2.3}$$

Esse modelo corresponde à quantificação multi-classe, que é uma representação generalizada da quantificação. Neste trabalho, é explorada a quantificação binária, na qual, há apenas duas classes de interesse, que podem ser definidas como positiva e negativa, ou, $\{0,1\}$. Tipicamente, o modelo busca estimar a prevalência da classe positiva, $\hat{p}$,

$$h : X^n \rightarrow [0,1] \tag{2.4}$$

A partir disso, a estimativa da prevalência da classe negativa também pode ser obtida:

$$\hat{n} = 1 - \hat{p} \tag{2.5}$$

2.2.2 Métodos de Quantificação

Os métodos de quantificação podem ser subdivididos em duas categorias principais: agregativo e não agregativo [6]. Os métodos agregativos são aqueles que requerem a classificação e os não agregativos são aqueles em que a quantificação é realizada sem a necessidade de classificação. Neste trabalho, o foco são os métodos agregativos, que são discutidos em mais detalhes.

2.2.2.1 Classify and Count (CC)

O CC [4] consiste em três passos primordiais: gerar um classificador a partir de um conjunto de dados de treino, ou seja, um conjunto previamente categorizado; utilizar esse classificador para gerar as classes de um conjunto de dados de teste (não categorizado); e, por fim, estimar a prevalência no conjunto de teste realizando a contagem de cada classe estimada. Considerando um contexto de quantificação binária, seria o equivalente a contabilizar as classes positivas estimadas. Portanto, o objetivo do CC é calcular:

$$\hat{p}^{CC}_{\oplus}(U) = \frac{\sum_{x \in U} h_{\oplus}(x)}{|U|} \tag{2.6}$$

O CC é uma abordagem simples da quantificação, não há qualquer tipo de ajuste. Isso implica que o CC busca, em geral, minimizar medidas como $(FP + FN)$, onde FP são os falsos positivos e FN são os falsos negativos, utilizadas especificamente para a classificação. A quantificação, no entanto, busca minimizar medidas como $|FP - FN|$. Por essa razão, foram desenvolvidos diferentes métodos que buscam realizar ajustes no CC.

2.2.2.2 Adjusted Classify and Count (ACC)

O ACC [4, 15] realiza correções nas estimativas fornecidas pelo CC, a partir das taxas de verdadeiros positivos ou *True Positive Rate* (*TPR*) e falsos positivos ou *False Positive Rate* (*FPR*), dadas por:

$$tpr = \frac{TP}{TP+FN} \qquad fpr = \frac{FP}{TN+FP} \tag{2.7}$$

Para o caso binário, a prevalência real de um conjunto de dados pode ser descrita da seguinte forma:

$$\hat{p}_{\oplus}(U) = \frac{\hat{p}^{CC}_{\oplus}(U) - fpr_h}{tpr_h - fpr_h} \tag{2.8}$$

sendo as taxas definidas como:

$$\hat{tpr} = \frac{\sum_{x\in\oplus} h_{\oplus}(x)}{|\oplus|} \qquad \hat{fpr} = \frac{\sum_{x\in\ominus} h_{\oplus}(x)}{|\ominus|} \tag{2.9}$$

A partir disso é possível obter a prevalência estimada pelo ACC, $\hat{p}^{ACC}_{\oplus}$, substituindo as taxas estimadas com os dados de treino:

$$\hat{p}^{ACC}_{\oplus}(U) = \frac{\hat{p}^{CC}_{\oplus}(U) - \hat{fpr}_h}{\hat{tpr}_h - \hat{fpr}_h} \tag{2.10}$$

2.2.2.3 Probabilistic Classify and Count (PCC)

O PCC [2] é uma variante do CC que estima as prevalências, a partir da contagem das probabilidades de uma dada classe. Esse método considera que as probabilidades posteriores trazem informações mais relevantes do que uma decisão binária. De maneira semelhante ao CC, o PCC pode ser definido da seguinte forma:

$$\hat{p}^{PCC}_{\oplus}(U) = \frac{\sum_{x\in U} s_{\oplus}(x)}{|U|} \tag{2.11}$$

A diferença entre os dois é que o PCC utiliza um classificador *soft*, enquanto o CC utiliza uma classificador *hard*, ou seja, o PCC utiliza um classificador que retorna as probabilidades posteriores, enquanto o CC retorna uma decisão binária.

2.2.2.4 Probabilistic Adjusted Classify and Count (PACC)

O PACC [2] é uma variante do CC que combina o ACC e o PCC, dessa forma, com base nas relações definidas em 2.10 e 2.11:

$$\hat{p}_{\oplus}^{PACC}(U) = \frac{\hat{p}_{\oplus}^{ACC}(U) - f\hat{p}r_s}{t\hat{p}r_s - f\hat{p}r_s} \tag{2.12}$$

2.2.3 Métricas de Avaliação

Como forma de avaliar os métodos de quantificação, foram utilizadas duas medidas, sendo elas o *Absolute Error (AE)* e o *Relative Absolute Error (RAE)*. Essas métricas são definidas formalmente da seguinte forma:

$$AE(p,\hat{p}) = \frac{1}{|Y|}\sum_{y\in Y}\left|\hat{p}_y - p_y\right| \qquad RAE(p,\hat{p}) = \frac{1}{|Y|}\sum_{y\in Y}\frac{\left|\hat{p}_y - p_y\right|}{p_y} \tag{2.13}$$

na qual Y representa o conjunto das classes de interesse.

Como o trabalho também consiste no desenvolvimento de um método com bom desempenho também para a classificação, foram utilizadas a acurácia (*accuracy*) e o *f1-score*. A acurácia pode ser definida como:

$$acc = \frac{TP+TN}{TP+FP+TN+FN} \tag{2.14}$$

E o *f1-score* corresponde à média harmônica entre precisão $TP/(TP+FP)$ e *recall*, $TP/(TP+FN)$:

$$f1 = \left(\frac{Recall^{-1} + Precision^{-1}}{2}\right) \tag{2.15}$$

2.2.4 BERT

Esta seção apresenta uma visão geral do Bidirectional Encoder Representations from Transformers (BERT), o classificador utilizado nos experimentos deste trabalho. Além disso serão abordados alguns conceitos relacionados ao classificador.

2.2.4.1 Word Embeddings

Em geral, os modelos de aprendizado de máquina utilizam vetores, contendo números, como seu argumento de entrada. Isso, no entanto, não pode ser aplicado de maneira direta a tarefas de Processamento de Linguagem Natural (PLN) ou *Natural Language Processing* (NLP), uma vez que os dados estão representados em forma de texto, que requer uma etapa de estruturação [1]. Sendo assim, há a necessidade de utilizar uma estratégia que realize uma

conversão de *strings* para vetores de características, para que os modelos possam ser treinados de forma adequada.

As *Word embeddings* [17] são uma dessas estratégias, na qual é fornecida uma representação em que palavras similares possuem uma codificação semelhante. Elas constituem um conjunto de métodos nos quais palavras individuais são representadas como vetores de números reais densamente distribuídos em um espaço vetorial [11]. Essa representação distribuída é atingida tomando como base que palavras utilizadas em contextos semelhantes possuam representações semelhantes, capturando o seu sentido de forma mais semântica.

As *Word Embeddings* se diferenciam de outros métodos por sua eficiência computacional. Elas utilizam vetores de dezenas ou centenas de dimensões, contrastando com os milhares ou milhões de dimensões utilizados em representações esparsas, como as clássicas *Bag-of-Words* [1].

2.2.4.2 Transformers

Transformers é o nome de um modelo de *deep learning*, muito utilizado no campo de processamento de linguagem natural, que adota o mecanismo de atenção [16], determinando a influência de diferentes partes dos dados de entrada. Ele é utilizado em tarefas como traduções, sumarização de textos, reconhecimento de falas, entre outros.

Assim como os *Transformers*, os modelos *sequence-to-sequence* baseados em redes neurais recorrentes são amplamente difundidos no campo de PLN. O objetivo desses modelos é converter uma sequência para outra sequência. Para tanto, o *sequence-to-sequence* faz uma análise sequencial, tomando como base o conjunto dos dados já percorrido para compreender o contexto da sequência como um todo [17]. Devido a essa natureza, esses modelos possuem limites no processamento do fim de uma sequência antes do início, bem como limites de paralelização, dificultando realizar tarefas com longas sequências.

Por outro lado, os *Transformers* operam de modo que, para uma dada sequência, é analisado a relevância que cada elemento possui. Sendo assim, o mecanismo de atenção examina a sequência de entrada e determina para cada etapa quais partes são as mais importantes. Como *Transformers* observam a sequência de maneira holística, não existe uma limitação para paralelização desta operação. No entanto, uma limitação dos *Transformers* é o fato de que o mecanismo de atenção requer sequências de tamanho fixo. Sendo assim, um texto grande deve ser repartido em partes menores, o que pode acarretar numa fragmentação do seu contexto.

2.2.4.3 Pré-treinamento

O BERT é um modelo de linguagem pré-treinado, ou seja, ele é treinado previamente utilizando uma grande quantidade de textos não anotados, o que permite que o modelo aprenda a linguagem em um contexto generalizado, para, posteriormente, ser treinado com um conjunto de

dados menor, específico para a tarefa que ele está realizando [3]. O PLN é um campo bastante diversificado, e, por isso, a quantidade de dados disponíveis e anotados para treino em uma tarefa específica, em geral, não é grande o suficiente. Os modelos de PLN demandam uma grande quantidade de dados para produzir resultados satisfatórios, o que motivou o BERT a realizar um pré-treinamento não supervisionado consistindo em duas tarefas.

A primeira delas acontece utilizando máscaras de modelos de linguagem modificadas, chamadas *Masked Language Model*, usando uma estratégia de processamento bidirecional. Ao contrário dos modelos de linguagem condicional que realizam um treinamento unidirecional, onde a palavra prevista é posicionada no fim ou no começo de uma sequência de texto, o BERT oculta uma palavra aleatória na sequência. Quando o modelo oculta uma palavra, ele substitui a palavra com um *token* especial. O modelo depois tenta prever a palavra ocultada utilizando o contexto dos lados direito e esquerdo com o auxílio de redes *Transformers*.

A segunda tarefa é a *Next Sentence Prediction*, que é utilizada como forma de entender a relação entre duas sentenças. O modelo recebe pares de frases como entrada e aprende a prever se a segunda frase do par é a frase subsequente no documento original. Essa tarefa, portanto, assume que frases subsequentes são conexas, e frases em posições aleatórias possuem menos relação.

2.2.4.4 Fine Tuning

O *fine-tuning* do BERT, ou seja, quando ele é ajustado para uma aplicação real em uma tarefa específica, acontece de maneira relativamente simples, adicionando apenas mais uma camada no modelo. Por isso, é computacionalmente muito menos custoso do que o pré-treinamento. Nesta fase, a maioria dos hiper-parâmetros se mantém os mesmos e o modelo se ajusta com base na tarefa a ser realizada, podendo ser uma classificação, resposta a perguntas ou reconhecimento de entidades nomeadas (*NER*).

No caso de análise de sentimentos, é realizado um *fine-tuning* adicionando uma camada extra ao BERT. Essa camada é responsável por classificar a polaridade (positiva ou negativa) dos textos, mantendo o restante dos parâmetros do BERT inalterados. Na prática, essa estratégia também pode ser considerada como uma transferência de aprendizado, uma vez que o modelo pré-treinado já apresenta representações vetoriais dos textos com o entendimento geral da linguagem.

Capítulo 3

DESENVOLVIMENTO

3.1 Considerações Iniciais

Neste capítulo, são apresentados os detalhes do desenvolvimento do projeto. O capítulo inicia-se com a descrição do problema abordado, onde é apresentada uma visão geral da metodologia aplicada. Na sequência, as etapas percorridas para a obtenção dos resultados são detalhadas. No capítulo seguinte, os resultados, divididos entre quantificação e classificação, são apresentados e discutidos.

3.2 Descrição do Projeto

Com o intuito de desenvolver um modelo capaz de apresentar resultados comparáveis ao estado da arte na quantificação aplicada a análise textual, este trabalho adapta o framework de quantificação proposto por Moreo e Sebastiani (2021) [12] para incluir um classificador baseado no BERT para quantificação CC. Nesse contexto, foi aplicada uma otimização para seleção de parâmetros do modelo baseada nas medidas de erro AE e RAE, próprias para a quantificação.

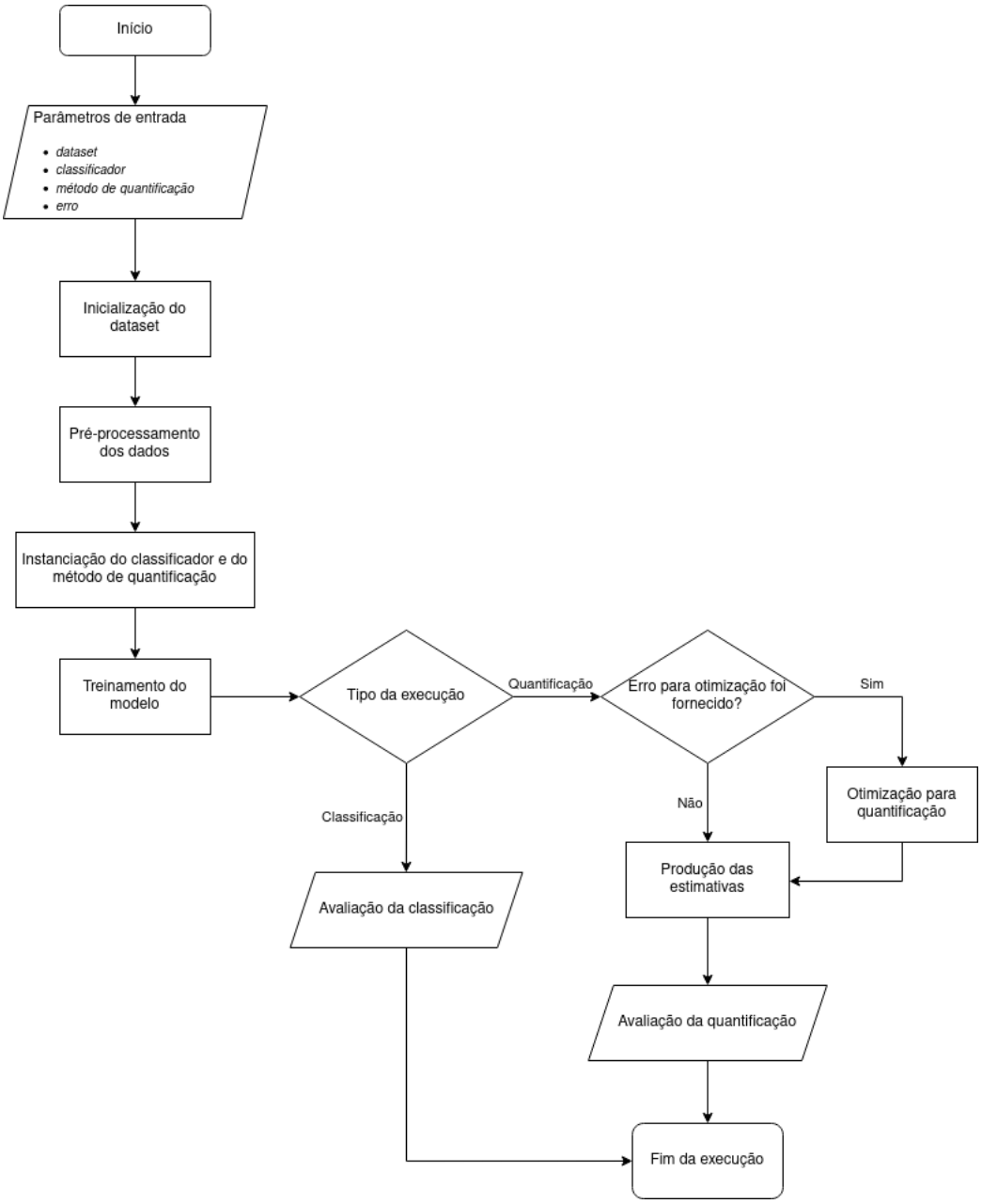
A ideia geral é treinar inicialmente o BERT para o problema de classificação de sentimentos. Dado um modelo treinado, é proposto o ajuste de seus parâmetros de classificação (na etapa de teste) considerando medidas de erro de quantificação. Esse ajuste é baseado nos limiares de confiança de classificação do BERT, que será discutido a seguir.

A implementação da aplicação foi inteiramente realizada em *Python*. Foi utilizado como base o código apresentado no trabalho de Moreo e Sebastiani (2021) [12]. Para incorporar o BERT ao framework, foi utilizada a biblioteca *Keras*, uma biblioteca de *deep learning* open-source, que atua como uma interface para a biblioteca *TensorFlow*.

3.3 Atividades Realizadas

Nesta seção, é detalhada a metodologia aplicada para obtenção dos resultados. O diagrama da Figura 1 apresenta o fluxo de informação na aplicação, as etapas são detalhadas em seguida.

Figura 1 – Diagrama representado o fluxo de informações para obtenção os resultados



3.3.1 Parâmetros de entrada

Para o início da execução, há três argumentos que, obrigatoriamente, devem ser fornecidos. O primeiro deles é o conjunto de dados (*dataset*) a ser analisado. Neste trabalho, três *datasets* de *benchmark* estão disponíveis: *kindle*, um conjunto de avaliações do leitor de livros digitais *Kindle*; *imdb*, um conjunto de avaliações de filmes do site *IMDB*; e *hp*, um conjunto de avaliações da série de livros *Harry Potter*. A Tabela 1 detalha a distribuição de classes em cada um dos *datasets*. Como é possível observar, o *IMDB* é um conjunto balanceado, apresentando prevalências semelhantes entre classes positivas e negativas, ao contrário do *kindle* e *hp*, que apresentam um desbalanceamento significativo, com uma grande prevalência de classes positivas.

O segundo argumento que deve ser fornecido é o classificador. Para que seja executado o

Tabela 1 – Prevalências representadas pelas proporções das classes positivo e negativo e totais de documentos por *dataset*

	$\oplus$	$\ominus$	Total de documentos para treino	Total de documentos para teste
kindle	0,917	0,083	3821	21592
imdb	0,500	0,500	25000	25000
hp	0,982	0,018	9533	18401

experimento proposto neste trabalho, o classificador especificado deverá ser o BERT.

O terceiro argumento é o método de quantificação. O método de quantificação adaptado para incorporar o *BERT* é o *Classify and Count* (CC).

Por fim, há um argumento opcional, que é a medida de erro a ser utilizada para otimizar parâmetros do modelo em uma tarefa de quantificação. Sendo assim, para realizar as otimizações, deverá ser inserido um tipo de medida de erro, podendo ser tanto erros voltados à classificação, como a acurácia (*acce*), dada por $(TP+TN)/TOTAL$; e *f1-score* (*f1e*), que corresponde à média harmônica entre precisão e *recall*; quanto erros voltados à quantificação como o *Absolute Error* (*AE*) e o *Relative Absolute Error* (*RAE*). Os termos apresentados entre parênteses correspondem à notação utilizada no código.

3.3.2 Inicialização do dataset, pré-processamento e instanciação do classificador e do método de quantificação

Assim que é iniciada a execução, o *dataset* escolhido é carregado para a memória. Esse *dataset*, no entanto, ainda deve passar por um pré-processamento. Como tratam-se de dados textuais, é necessária que haja uma adaptação para que o classificador possa manipular esses dados, e a função deste pré-processamento é converter todo o *dataset* para uma forma que possa ser compreendida pelo BERT.

Como os dados são inicializados na forma de vetores, para realizar o pré-processamento, foi utilizado método *texts_from_array*, as classes então são definidas como 0 e 1, representando negativo e positivo. Devido à natureza das redes *Transformers*, é necessário definir um tamanho limite para os documentos. No experimento realizado foi definido um tamanho de 128 tokens por documento, mas que podem ser ajustado para o limite de 512 tokens. Feito o pré-processamento dos dados, é definido um modelo classificador baseado no BERT.

Nesta fase, portanto, foram inicializados os dados pré-processados, o modelo e o classificador. Todos eles serão utilizados para instanciar um objeto do classificador BERT. Por fim, o método de quantificação fornecido é instanciado, incorporando o classificador.

3.3.3 Treinamento do modelo

O BERT foi treinado utilizando o método *fit_onecycle* [14]. Como parâmetros, o *learning rate* foi especificado em 5×10^{-5} e três iterações (*epochs*) foram definidas, uma vez que obtiveram bons resultados de classificação em um conjunto de validação.

3.3.4 Avaliação da classificação

Caso a execução tenha a finalidade de avaliar o classificador para o *dataset* especificado, após o treinamento do modelo, serão realizadas predições com o conjunto de teste, e a partir dos resultados, serão extraídas as métricas de classificação.

3.3.5 Otimização para quantificação

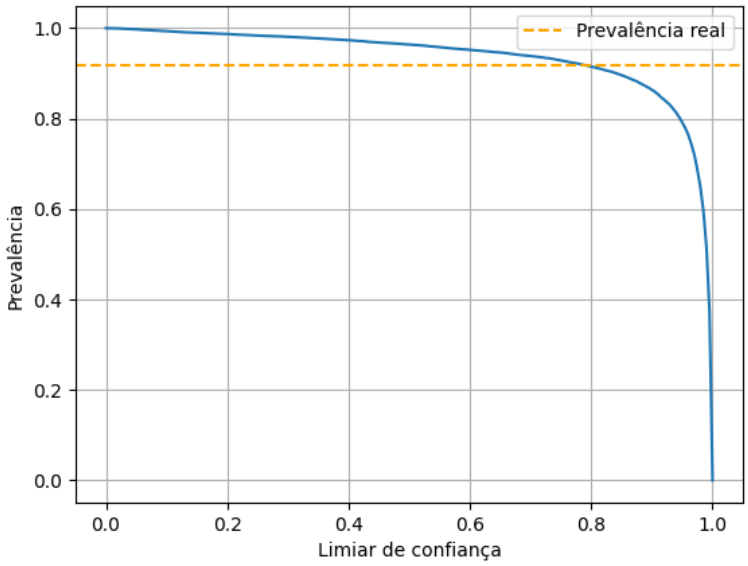
Diferentemente dos demais classificadores presentes no código, apenas um modelo do BERT será treinado. Para o restante dos classificadores, é realizada uma série de treinamentos variando os parâmetros, e, com base na métrica de erro fornecida, é selecionado o modelo com a melhor avaliação.

Em todos os experimentos envolvendo otimização, o *dataset* de treino foi dividido em uma proporção de 60% de dados para treino e 40% de dados para validação, mantendo o padrão do experimento base [12]. Os dados de validação são usados para analisar parâmetros que otimizam o modelo para a quantificação.

Para o BERT, foi adotada uma estratégia de otimização diferente. Primeiramente, é realizado um único treinamento e, a partir do classificador obtido, são estimadas as probabilidades posteriores sob os dados de validação. O limiar de confiança que determina se um texto será positivo ou negativo, caso não houvesse uma otimização, seria 0,5. O objetivo dessa otimização, portanto, será encontrar o melhor limiar, de modo a definir um modelo que maximize os resultados para a tarefa de quantificação. Sendo assim, a partir dos dados de treino e validação, são realizados múltiplos testes com diferentes limiares. Aquele que obtiver o melhor resultado é selecionado, e é adicionado como parâmetro do modelo.

A lógica desta abordagem reside no fato de que nos conjuntos de dados não balanceados um limiar de confiança de $0,5$ não produz os melhores resultados para quantificação, embora possa obter resultados promissores para classificação. Uma simulação realizada com o *dataset* não balanceado *kindle*, na qual foram obtidas as probabilidades posteriores e determinada a prevalência da classe positiva para diferentes limiares a partir do método CC, permite observar a variação da prevalência. O gráfico dos resultados é apresentado na Figura 2.

Figura 2 – Gráfico das prevalências obtidas para diferentes limiares de confiança



3.3.6 Produção das estimativas e Avaliação da quantificação

Após o treinamento e seleção do melhor limiar de confiança para quantificação, o modelo produz as estimativas para os dados de teste. Para realizar as estimativas, a aplicação subdivide os dados de teste por prevalências, de modo a realizar uma série de testes para conjunto de dados com características distintas. Por padrão, são determinados 21 níveis de prevalência, ou seja, são realizadas estimativas 21 vezes para conjuntos de dados em que a prevalência da classe positiva vai de 0 a 1, com intervalos de 0,05. Os resultados, então, seguem para uma avaliação, onde serão calculadas as medidas de erro para quantificação.

3.4 Dificuldades e Limitações

A principal dificuldade encontrada no desenvolvimento do projeto foi a limitação na capacidade de processamento. Devido ao grande volume de dados trabalhados e o gasto computacional que o BERT requer, não foi possível executar os experimentos em máquinas pessoais.

Para isso foram usados serviços em nuvem que, apesar de serem capazes de executar o código, possuem restrições quanto ao tempo de sessão e de utilização da GPU.

Capítulo 4

RESULTADOS

4.1 Avaliação da Quantificação

O trabalho de Moreo e Sebastiani (2021) [12] traz um experimento que contempla 5 tipos diferentes de classificadores, sendo eles: SVM (Máquina de Vetores de Suporte), LR (Regressão Logística), RF (*Random Forest*, MNB (*Naive Bayes* Multinomial e CNN (Rede neural convolucional). Portanto, o objetivo ao realizar os experimentos com o método CC utilizando o BERT é compará-lo com os demais classificadores. Os testes foram realizados nos mesmos conjuntos de dados, a fim de determinar qual o impacto que se obtém ao utilizar o BERT, incorporando o protocolo de otimização para uma tarefa de quantificação.

Como forma de avaliar os resultados, foram escolhidos o *Absolute Error* (AE) e o *Relative Absolute Error* (RAE), duas métricas específicas para quantificação.

Foram realizadas duas simulações para cada conjunto de dados. Na primeira delas, o modelo não é otimizado para a tarefa de quantificação, ou seja, é o classificador CC tradicionalmente adotado na literatura. Na segunda simulação, foi determinado o AE como medida de erro, ou seja, foi realizada uma otimização para seleção de modelos para tarefa de quantificação CC, conforme sugerido por Moreo e Sebastiani (2021) [12].

Os resultados obtidos no experimento de quantificação, comparados com os demais classificadores, podem ser observados na Tabela 2. O símbolo ∅ indica que nenhuma métrica de erro foi especificada e, portanto, nenhuma otimização foi realizada, enquanto as linhas marcadas com "AE"indicam a realização de otimização com base no erro absoluto. Além disso, os limiares de confiança definidos nas otimizações do BERT para quantificação estão presentes na Tabela 3.

Como pode ser observado, os resultados para o BERT quando comparados com os demais métodos são competitivos, sendo que em quatro das seis medidas realizadas foi ele que apresentou os melhores resultados.

Analisando os dados, fica clara a influência que os *datasets* tem nos resultados. Para o *IMDB*, um conjunto balanceado, foram obtidos os melhores resultados. Entretanto, a otimização neste caso surtiu pouco ou nenhum efeito, visto que para o RAE houve uma melhoria de aproximadamente 0,3%, enquanto que, para o AE, nenhuma melhoria foi observada. Essa pouca influência da otimização também pode ser observada na Tabela 3. O limiar obtido neste caso foi de 0,5772, pouco acima do caso padrão para classificação, no qual 0,5 é o limiar.

Tabela 2 – Resultados para o método CC para os três *datasets* disponíveis, indicando a realização de otimização com base no *Absolute Error* (AE) ou a não realização de otimização (∅). Foram extraídas as métricas AE e RAE para avaliar a quantificação. Os resultados para SVM, LR, RF, MNB e CNN foram extraídos do trabalho de Moreo e Sebastiani (2020) [12]

		IMDB		**KINDLE**		**HP**	
Classificador	**Erro**	*AE*	RAE	AE	RAE	AE	RAE
SVM	∅	0,065	6,029	0,305	15,928	0,471	24,058
	AE	0,065	6,091	0,100	7,555	0,119	10,593
LR	∅	0,059	5,477	0,470	23,990	0,500	25,508
	AE	0,062	5,745	0,094	7,087	**0,110**	10,304
RF	∅	0,155	13,388	0,448	22,988	0,493	25,196
	AE	0,079	7,487	0,464	23,721	0,500	25,487
MNB	∅	0,096	8,147	0,500	25,513	0,500	25,510
	AE	0,097	8,431	0,443	22,701	0,499	25,464
CNN	∅	0,072	6,683	**0,087**	8,138	0,255	17,042
	AE	0,074	6,613	0,106	8,591	0,343	19,008
BERT	∅	**0,056**	5,163	0,148	8,113	0,295	15,345
	AE	**0,056**	**5,146**	0,103	**6,204**	0,122	**7,913**

Tabela 3 – Limiares de confiança obtidos nas otimizações para os três *datasets*.

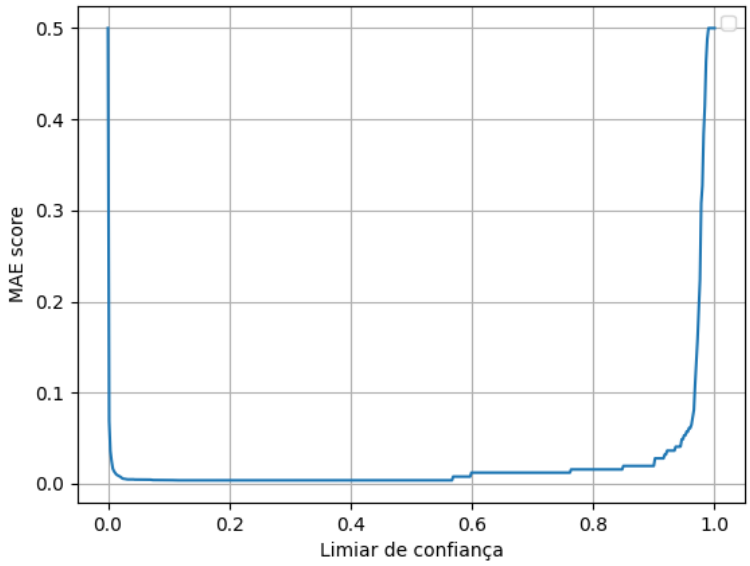
Dataset	Limiar de confiança
IMDB	0,5772
KINDLE	0,1142
HP	0,0240

Uma grande diferença, em relação ao limiar de confiança, pode ser notada para o *dataset Kindle*, que possui um desbalanceamento a favor da classe positiva. Nesse cenário, a otimização realizou uma mudança brusca no limiar, que foi estabelecido em 0,1142. A influência da otimização é notada também na diferença entre as avaliações sem medida de erro definida e com o AE definido. Foi obtida uma melhoria de 30,4% no AE e de 23,5% para o RAE, demonstrando a necessidade de otimizar o BERT para quantificação.

Por fim, no *dataset HP*, um conjunto extremamente desbalanceado em favor da classe positiva, observou-se uma grande melhoria ao realizar-se a otimização, alcançando uma melhoria de 58,6% para o AE e de 48,4% para o RAE. O limiar de confiança também sofreu uma grande alteração, chegando a 0,0240. A Figura 3 mostra a variação do AE durante a otimização, é possível observar que, a partir do limiar de confiança 0,1142, obtém-se os melhores resultados.

Uma análise conjunta dos dados apresentados nesta seção comprovam o benefício trazido por um classificador estado-da-arte como o BERT para uma tarefa de quantificação, utilizando o CC, o método de quantificação trivial. Além disso, outro fato observado é a influência que uma otimização, a partir da variação do limiar de confiança, provoca em conjunto de dados não balanceados.

Figura 3 – Gráfico do erro absoluto obtido para diferentes limiares de confiança no *dataset Kindle*



4.2 Avaliação da Classificação

Uma segunda forma de analisar o experimento como um todo, foi a avaliação dos classificadores em sua tarefa de origem. Como a proposta deste trabalho é a aplicação de um método que está intrinsecamente relacionado à classificação, é importante haver uma avaliação que comprove a eficiência do método proposto tanto para tarefas de quantificação quanto de classificação.

Sendo assim, foi realizada mais uma simulação para cada um dos classificadores, em cada um dos três *datasets*. Desta vez, os modelos foram treinados normalmente, mas não houve qualquer tipo de otimização. Após o treinamento, os modelos foram testados e as métricas de avaliação coletadas. Para os classificadores SVM , LR, RF, MNB e CNN foi utilizado o método *classification_report* da biblioteca *scikit-learn*. Para o BERT foi utilizado o método *evaluate* para classificadores da biblioteca *ktrain*. Ambos retornam as principais métricas para classificação, e, para analisar os resultados obtidos, foram selecionadas a acurácia e o *F1-Score*. O valor de *F1-Score* corresponde a uma média entre os valores calculados para a classe positiva e a classe negativa. Os resultados podem ser observados na Tabela 4.

Novamente, é possível concluir que a aplicação do BERT traz grandes ganhos ao realizar a classificação, uma vez que para as duas métricas analisadas, o BERT apresentou os melhores resultados em todos os conjuntos de dados.

Tabela 4 – Métricas de avaliação obtidas para os seis classificadores, envolvendo os três datasets. ACC indica a acurácia e F1 indica o F1-Score.

	IMDB		**KINDLE**		**HP**	
Classificador	ACC	F1	ACC	F1	ACC	F1
SVM	0,88	0,88	0,94	0,50	0,94	0,54
LR	0,89	0,89	0,94	0,54	0,94	0,48
RF	0,84	0,84	0,94	0,50	0,94	0,48
MNB	0,83	0,83	0,94	0,48	0,94	0,48
CNN	0,86	0,86	0,92	0,74	0,89	0,61
BERT	**0,90**	**0,90**	**0,97**	**0,86**	**0,96**	**0,79**

4.3 Considerações Finais

Analisando isoladamente alguns métodos, como o CNN, é possível observar que, apesar de ele ter apresentado bons resultados em relação ao F1-Score na classificação, o mesmo não acontece com a acurácia. Além disso, no caso da quantificação, observa-se resultados competitivos no *dataset Kindle*, porém, nos demais conjuntos ele não teve destaque.

Dessa forma, é interessante destacar a consistência apresentada pela proposta, na qual o BERT apresentou resultados satisfatórios, mantendo-se competitivo em todos os cenários analisados.

Por fim, é possível concluir que a ideia inicial deste trabalho, de desenvolver um método capaz de ter bons resultados em duas tarefas distintas realizando um único treinamento provou-se válida. As implicações práticas dessa característica são bastante interessantes, visto a demanda computacional que os classificadores requerem nas fases de treinamento e otimização. E, por isso, vale destacar que o estudo realizado neste trabalho viabiliza reuso de modelos em tarefas de classificação e quantificação de análise de sentimentos, o que é uma características desejada nos *pipelines* modernos de ciência de dados.

O código-fonte do projeto está disponível em: https://github.com/gbnogima/CC

Capítulo 5

CONCLUSÃO

5.1 Contribuições

Em relação ao trabalho realizado, é interessante observar, por meio de experimentos, o impacto proporcionado por um classificador robusto como o BERT. Os resultados foram satisfatórios e conseguiram alcançar a expectativa inicial de desenvolver um modelo que, a partir de um único treinamento, apresente resultados competitivos na classificação e quantificação.

A nível de contribuição, vale ressaltar o conhecimento prático adquirido em campos como a análise de sentimento, aprendizado de máquina e análise de dados em geral, áreas de bastante relevância atualmente. Além disso, a experiência de construir um trabalho acadêmico ao longo de um semestre permite compreender diversos aspectos sobre o tópico estudado, proporcionando uma visão aprofundada do problema.

REFERÊNCIAS

[1] C. C. Aggarwal. *Machine learning for text*. Springer, 2018. Citado 3 vezes nas páginas 21, 24 e 25.

[2] A. Bella, C. Ferri, J. Hernández-Orallo, and M. J. Ramirez-Quintana. Quantification via probability estimators. In *2010 IEEE International Conference on Data Mining*, pages 737–742. IEEE, 2010. Citado 2 vezes nas páginas 23 e 24.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. Citado 2 vezes nas páginas 18 e 26.

[4] G. Forman. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206, 2008. Citado 2 vezes nas páginas 22 e 23.

[5] Z. Gao, A. Feng, X. Song, and X. Wu. Target-dependent sentiment classification with bert. *IEEE Access*, 7:154290–154299, 2019. Citado na página 18.

[6] P. González, A. Castaño, N. V. Chawla, and J. J. D. Coz. A review on quantification learning. *ACM Computing Surveys (CSUR)*, 50(5):1–40, 2017. Citado 2 vezes nas páginas 17 e 22.

[7] P. González, J. Díez, N. Chawla, and J. J. del Coz. Why is quantification an interesting learning problem? *Progress in Artificial Intelligence*, 6(1):53–58, 2017. Citado na página 17.

[8] M. Hoang, O. A. Bihorac, and J. Rouces. Aspect-based sentiment analysis using bert. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 187–196, 2019. Citado na página 18.

[9] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer, 2012. Citado na página 18.

[10] A. Maletzke, D. dos Reis, E. Cherman, and G. Batista. Dys: a framework for mixture models in quantification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4552–4560, 2019. Citado na página 18.

[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 3111–3119, 2013. Citado na página 25.

[12] A. Moreo and F. Sebastiani. Re-assessing the "classify and count" quantification method. In D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, and F. Sebastiani, editors, *Advances in Information Retrieval*, pages 75–91, Cham, 2021. Springer International Publishing. ISBN 978-3-030-72240-1. Citado 8 vezes nas páginas 9, 17, 18, 19, 27, 30, 33 e 34.

[13] B. N. Santos, R. M. Marcacini, and S. O. Rezende. Multi-domain aspect extraction using bidirectional encoder representations from transformers. *IEEE Access*, 2021. Citado na página 18.

[14] L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017. Citado na página 30.

[15] D. Tasche. Fisher consistency for prior probability shift. *The Journal of Machine Learning Research*, 18(1):3338–3369, 2017. Citado na página 23.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017. Citado na página 25.

[17] S. Wang, W. Zhou, and C. Jiang. A survey of word embeddings based on deep learning. *Computing*, 102(3):717–740, 2020. Citado na página 25.

[18] L. Zhang, S. Wang, and B. Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018. Citado na página 18.