

# Avaliação do uso de Transferência de Aprendizagem para Realização de Análise de Sentimentos no Sistema de Apoio à Participação em Eventos MyMobiConf

Lucas Gabriel B. Cunha<sup>1</sup>, Thais Regina de Moura Braga Silva<sup>2</sup>

<sup>1</sup>Instituto de Ciências Exatas e Tecnológicas – Universidade Federal de Viçosa (UFV)  
35.690-000 – Florestal – MG – Brasil

{lucas.g.cunha, thais.braga}@ufv.br

**Resumo.** A transferência de aprendizagem consiste em aproveitar modelos relacionados à solução de uma tarefa, para melhorar o aprendizado de outra, mostrando-se um tópico importante, uma vez que lida com problemas comumente enfrentados durante a construção de soluções de aprendizado de máquina, tais como a qualidade e a quantidade de dados. O objetivo desse estudo foi avaliar como o uso de técnicas de transferência de aprendizagem pode afetar, positiva ou negativamente, o modelo de análise de sentimentos presente no sistema de apoio à eventos MyMobiConf. Para tanto, foi implementado o algoritmo TrAdaBoost, que visa realizar a transferência de instâncias, e analisado o comportamento do modelo final gerado. Os resultados do trabalho mostraram que o uso do algoritmo selecionado pode ser benéfico, nos parâmetros apresentados, quando o número de iterações para o mesmo é de até 25 iterações, ocasionando em um aumento de até 9% na acurácia do modelo.

## 1. Introdução

Atualmente, muito tem se discutido sobre o uso do aprendizado de máquina e como sua utilização pode trazer benefícios para as mais diversas entidades. Um tópico constantemente abordado é a utilização de modelos de análise de sentimento, que visam realizar a classificação de textos de acordo com um treinamento recebido previamente, a fim de obter um melhor desempenho na tarefa designada [Medhat et al. 2014].

Modelos de aprendizado de máquina destinados à análise de sentimento, se bem treinados, podem trazer grandes benefícios para a entidade alvo, uma vez que ajudam a avaliar, de forma rápida, o sentimento expressado no texto recebido. Em um contexto empresarial, a análise de sentimentos pode proporcionar um melhor entendimento da visão do cliente em relação à empresa ou produto, fazendo com que seja possível tomar decisões rápidas e identificar pontos de melhoria com base no retorno dado pelos clientes, ação esta que pode se converter, até mesmo, em ganhos financeiros para a empresa.

Apesar de ser um tópico bastante trabalhado, ainda podem haver dificuldades na implementação de um bom modelo de análise de sentimentos, uma vez que este processo pode enfrentar desafios como a escassez de dados e o desbalanceamento deles. Uma das alternativas para estes cenários é a utilização de técnicas de transferência de aprendizagem, que possuem o potencial de melhorar o conjunto de dados de treino do modelo por meio da utilização de dados externos que, apesar de também serem usados para o mesmo fim, pertencem a um domínio diferente do domínio visado [Pan and Yang 2010].

O MyMobiConf<sup>1</sup> é um sistema de apoio à eventos presenciais e online que faz uso de um modelo de análise de sentimentos, para que os organizadores possam ter uma visão geral da percepção de seu público para com o evento. O modelo em questão foi construído utilizando a base de dados do sistema, a qual enfrenta problemas quanto à quantidade de dados e ao balanceamento deles. Tal cenário, conseqüentemente, afeta a qualidade das métricas de avaliação do modelo, em especial das classificações relacionadas à opiniões negativas, o que é justificado pela escassez de registros de dados referentes a opiniões negativas na base de treino.

Tendo em vista as dificuldades de classificação enfrentadas pelo modelo de análise de sentimentos do MyMobiConf, o mesmo pode vir a ser beneficiado pela implementação de técnicas de transferência de aprendizagem, uma vez que sua base de treino se tornará mais robusta. Portanto, o objetivo deste trabalho é avaliar o uso de transferência de aprendizagem para o modelo de análise de sentimentos do sistema MyMobiConf, a fim de verificar se a utilização de tal abordagem pode trazer vantagens para as métricas finais de avaliação do modelo.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta alguns trabalhos relacionados encontrados na literatura. A Seção 3 descreve a fundamentação teórica, abordando os tópicos análise de sentimentos e transferência de aprendizagem. A Seção 4 introduz o sistema MyMobiConf e o atual modelo de análise de sentimentos, bem como trata da implementação da transferência de aprendizagem no modelo. A seção 5 apresenta os resultados dos cenários abordados e, por último, a seção 6 apresenta a conclusão da análise e os trabalhos futuros.

## 2. Trabalhos Relacionados

O trabalho [Pan and Yang 2010] explora os métodos tradicionais de análise de sentimentos e realiza a implementação de diferentes métodos de transferência de aprendizagem para *datasets*, atualmente famosos, utilizados para a construção de modelos de análise de sentimentos, chegando à conclusão de que a maior dificuldade na implementação da transferência de aprendizagem é conseguir lidar com a transferência negativa de aprendizagem, tópico que, apesar de importante, ainda não foi muito explorado.

Em [Dai et al. 2007] é introduzido um *framework* para a transferência de aprendizagem por meio do método de transferência de instâncias, em que a ideia principal é utilizar mecanismos de *boosting* para filtrar informações relevantes de uma base de dados externa. O estudo conclui que a utilização do algoritmo proposto gera melhores resultados do que técnicas tradicionais de aprendizado, além de resultar em uma performance superior a quase todos os cenários apresentados.

## 3. Fundamentação Teórica

### 3.1. Análise de Sentimentos

A Análise de Sentimentos (AS) é uma aplicação de processamento de linguagens naturais que visa, por meio da análise das palavras, definir a emoção expressada em um determinado texto. Em uma definição formal, a AS ou Mineração de Opinião (MO), como também pode ser chamada, é o estudo computacional da opinião, atitude e emoção de

---

<sup>1</sup><http://www.mymobiconf.caf.ufv.br/>

peças para com uma entidade, essa última, podendo representar indivíduos, eventos ou tópicos[Medhat et al. 2014].

Há várias formas de construir um modelo de análise de sentimentos, podendo ser por meio de aprendizados supervisionados ou não supervisionados, sendo possível a utilização de técnicas de aprendizado de máquina e também aprendizado profundo. Vários algoritmos podem ser utilizados para a criação do modelo, tais como:

- Random Forest;
- Recurrent Neural Network (RNN);
- Support Vector Machine (SVM);
- Naive Bayes.

O algoritmo deverá ser definido com base no problema a ser abordado, portanto, o entendimento do problema é fundamental para a construção do modelo, uma vez que cada um dos algoritmos possui sua própria peculiaridade.

Mesmo que a escolha do algoritmo seja importante, é fundamental construir um tratamento de dados robusto, principalmente lidando com palavras, uma vez que neste processo é possível seguir por diferentes caminhos, como a extração do radical das palavras (*stemming*), ou a remoção das inflexões da palavra, desde que resultem em uma palavra ainda existente (*lemmatization*), por exemplo. Deste modo, o processo de construção é constantemente cercado por tentativas e testes, a fim de decidir “o que” e “como” serão implementadas as etapas que farão parte da construção do modelo.

O modelo é, normalmente, criado para atender a um domínio específico e, portanto, é construído, na maioria das vezes, utilizando-se dados do domínio escolhido, para que o mesmo conheça os parâmetros utilizados e possua um melhor desempenho na tarefa designada.

Ademais, a quantidade e qualidade dos dados é fundamental para a criação de um bom modelo, e isso muitas vezes pode ser um desafio para os programadores, seja pela falta de abundância de amostras de treino, referentes ao domínio trabalhado, ou pela falta de diversidade entre as amostras, visto que, quanto maior o vocabulário utilizado para treino, mais o modelo estará preparado para lidar com cenários incomuns.

Tendo em vista as dificuldades provenientes da ausência de uma boa base de dados para o domínio desejado, maneiras de se contornar a falta de um conjunto adequado de informações foram desenvolvidas, a fim de reaproveitar dados de domínios externos, como é o caso da utilização de abordagens de transferência de aprendizagem.

### **3.2. Transferência de Aprendizagem**

Antes de definirmos o conceito de transferência de aprendizagem, é importante que seja definida a terminologia a ser utilizada:

- $D_s$  → O Domínio Fonte Contém o espaço do atributo de dados fonte, em outras palavras, contém os dados externos, os quais serão avaliados, a fim de se definir sua relevância para o treinamento do modelo referente ao Domínio alvo;
- $T_s$  → A Tarefa Fonte contém os atributos, referentes à  $D_s$ , os quais deseja-se prever;
- $D_t$  → O Domínio Alvo - O domínio ao qual a transferência de aprendizagem deve se basear;

- $T_t \rightarrow$  A Tarefa Alvo contém os atributos, referentes à  $D_t$ , os quais deseja-se prever.

Dito isso, transferência de aprendizagem pode ser definida como o processo de melhorar a função preditiva do alvo utilizando as informações de  $D_s$  e  $T_s$ , onde  $D_s \neq D_t$  e  $T_s \neq T_t$ .

Segundo [Pan and Yang 2010], as abordagens de transferência de aprendizagem podem ser divididas em quatro categorias: métodos de transferência de instância, métodos de transferência de parâmetro, métodos de transferência de representação de característica e métodos de transferência de aprendizagem relacional.

Uma vez que, cada categoria de transferência atende um determinado cenário, é necessário, portanto, antes de se decidir o algoritmo, identificar a qual das categorias o problema enfrentado pertence.

Na transferência de instâncias assume-se que, apesar dos dados de  $D_s$  não poderem ser diretamente utilizados para treino, pode ainda haver informações relevantes que possam ser utilizadas por  $D_t$  para que se fortaleça sua base de treino, portanto são utilizadas técnicas para que tais informações sejam “filtradas” e adicionadas à base. Abordagens de transferência de características, por sua vez, focam em minimizar a divergência entre domínios por meio de uma característica, considerada boa, que ambos possuam em comum. A transferência de parâmetros assume que modelos individuais que lidam com ações relacionadas devem possuir parâmetros em comum. Por último, a transferência de aprendizagem relacional lida com domínios relacionais, e busca transferir relações presentes entre os dados fonte para os dados alvo.

Apesar de abordagens de transferência de aprendizagem proverem bons caminhos para diferentes cenários, ainda há casos onde ela não gera um ganho de desempenho no modelo, podendo até mesmo reduzi-lo. A esses casos é dado o nome de transferência negativa de aprendizagem.

### 3.3. O Algoritmo TrAdaBoost

O algoritmo TrAdaBoost[Dai et al. 2007] é uma extensão do algoritmo AdaBoost[Freund and Schapire 1997] utilizado para transferência de aprendizagem de instâncias. Os dois algoritmos apresentam funcionamento similar uma vez que o TrAdaBoost também busca um melhor resultado por meio da combinação dos chamados *weak learners*, modelos que funcionam um pouco melhor que decisões aleatórias, e da repesagem das instâncias de treino com base no resultado do modelo gerado pela iteração anterior.

Na implementação do AdaBoost instâncias de treino preditas erroneamente pelo modelo recebem um maior peso para a próxima iteração, enquanto instâncias preditas corretamente tem seus pesos reduzidos (a soma dos pesos de todas as instâncias deve resultar em 1), assim o modelo sofrerá uma maior penalidade caso realize uma má classificação novamente, essa abordagem faz com que o modelo dê um maior enfoque em classificar corretamente instâncias com maiores pesos.

A cada nova iteração é criado um novo *dataset* de treino, o qual possui a mesma quantidade de registros do *dataset* inicial, porém seus registros são escolhidos aleatoriamente com base no peso de cada um dos registros da iteração anterior, aceitando

repetições. Portanto, registros que obtiveram um maior peso na iteração anterior tendem a aparecer um maior número de vezes no novo *dataset*, sendo assim, terão uma maior relevância para o treino e, desse modo, o modelo terá mais recursos para classificá-lo corretamente.

No TrAdaBoost instâncias de treino são divididas em dois grupos, instâncias fonte e instâncias alvo. As chamadas instâncias alvo fazem parte do domínio ao qual o modelo atuará, porém não possuem registros suficientes para treinar o modelo sozinho. Por outro lado, as instâncias fonte são encontradas em abundância apesar de pertencerem a outro domínio. Um modelo treinado apenas com registros do domínio fonte alcançaria um mal desempenho prevendo registros do domínio alvo. O TrAdaBoost é então utilizado para filtrar registros do domínio fonte que possam ser relevantes para o treinamento do modelo no domínio alvo. Portanto, o objetivo do TrAdaBoost é aumentar a base de treino final do modelo concatenando os registros da instância alvo com os registros julgados relevantes da instância fonte.

Dito isso, o TrAdaBoost realiza o chamado *reversing boosting*, em que, instâncias de treino, pertencentes ao domínio fonte mal preditas têm seu peso diminuído, uma vez que é entendido que tal registro poderá interferir negativamente no treinamento do modelo. Caso o processo de redução do peso não ocorresse, as instâncias poderiam ocasionar *overfitting*, que consiste, basicamente, do modelo acabar utilizando dados de ruído para seu treinamento, o que ocasiona uma queda no seu desempenho. Em contrapartida, para instâncias de treino mal preditas pertencentes ao domínio alvo é então utilizado o mesmo comportamento do AdaBoost em que o peso de tais instâncias é aumentado para que o modelo tenha uma maior preocupação em classificá-las corretamente. Tal comportamento difere ao que ocorre no AdaBoost, tendo em vista que, enquanto o AdaBoost busca prever corretamente os registros mal preditos durante as iterações o TrAdaBoost faz a diferenciação entre instâncias e diminui o peso de predições incorretas da instância fonte, a fim de tentar descartar sua relevância para o treinamento do modelo. De maneira didática, o TrAdaBoost, com esse comportamento, busca eliminar o registro da base de treino final, uma vez que o algoritmo definiu que ele não contribuirá positivamente para o treinamento. Outro ponto importante é que, enquanto o AdaBoost calcula o peso e erro não fazendo diferenciação entre instâncias, no TrAdaBoost o peso é calculado com base no somatório dos pesos totais para as instâncias fonte e para as instâncias alvo, tal que o resultado seja 1.

Uma vez claras as diferenças entre os algoritmos, o fluxo de funcionamento de ambos é similar. Inicialmente ocorre a normalização dos pesos das instâncias, em que, o somatório dos pesos de cada instância deve resultar em 1. Em seguida, um modelo é treinado utilizando as instâncias fonte e alvo, levando em consideração o peso de cada registro. Após o treinamento do modelo, é então calculado o erro total e os novos pesos das instâncias e este ciclo é seguido até que se atinja o número de iterações definidas.

## **4. Transferência de Aprendizagem para Análise de Sentimentos no Sistema MyMobiConf**

### **4.1. O Sistema MyMobiConf**

O MyMobiConf é um sistema de gerenciamento de eventos presenciais e online que possibilita que os participantes acompanhem as atividades do evento e deem seu *feedback* em

relação ao mesmo em tempo real, e assim os organizadores coletam dados para melhorar o engajamento de seu público.

Atualmente a plataforma conta com duas versões, uma versão *mobile*, destinada aos usuários, que possibilita que eles inscrevam-se em eventos cadastrados, acompanhem sua agenda e participem de suas interações em tempo real. Há também uma versão Web, destinada aos organizadores do evento. Nela é possível fazer configurações abrangentes sobre o evento, como criar novas atividades, cadastrar patrocinadores, emitir notícias para os participantes, além de ser possível visualizar o engajamento e o *feedback* dos usuários por meio de questionários, opiniões e notas que estão associadas a uma determinada atividade.

As opiniões funcionam como um *feedback* direto aos organizadores. Por meio delas é estabelecido um canal de comunicação confiável para que organizadores possam entender como seus participantes se sentem em relação ao evento, fazendo com que seja possível tomar decisões rápidas para melhorá-lo.

Na versão Web da aplicação, é possível visualizar todas as opiniões enviadas durante o evento, bastando acessar o evento por meio do painel de controle e em seguida acessar a página “Ver Opiniões” contida na seção de *feedbacks*. A partir da nova tela é possível alcançar dois tipos de visualizações, o primeiro utilizando do “Filtro de opiniões”, que permite que os organizadores busquem determinadas quantidades de opiniões com base em palavras chave inseridas. A segunda opção pode ser acessada pelo uso da aba “Estatísticas de opiniões” que disponibiliza métricas como:

- Total de opiniões enviadas;
- Participantes cadastrados no evento;
- Média de opiniões por participante;
- Participante que mais opinou;
- Participante que menos opinou;

Apesar das diferentes formas de visualização, tais abordagens não apresentam um *feedback* assertivo e direto sobre a qualidade do evento, uma vez que, em grandes eventos, por conta do grande fluxo de opiniões, não seria possível chegar a uma conclusão sobre a qualidade deles, apenas com as métricas apresentadas sem que fosse necessário ler os *feedbacks* individualmente.

Portanto, para facilitar o acompanhamento da qualidade do evento, foi implementado um modelo de análise de sentimentos, o qual prevê o sentimento para cada uma das opiniões inseridas e as classifica como positivas, negativas ou neutras. Com esse novo “agrupamento” torna-se possível que os organizadores filtrem opiniões com base em sua classificação feita pelo modelo, e assim possam identificar os pontos fracos do evento, apontados pelos participantes, por meio das opiniões definidas como negativas.

#### **4.2. A Análise de Sentimentos no Sistema MyMobiConf**

No MyMobiConf usuários participantes podem expressar opiniões sobre o evento por meio da utilização da aplicação *mobile*. As opiniões são, então, armazenadas no banco de dados da aplicação web, para que possam ser visualizadas pelos organizadores. As novas opiniões inseridas passam então por um processo de limpeza e vetorização para que então tenham sua classificação predita pelo modelo de análise de sentimentos implementado no sistema.

O modelo utiliza o algoritmo Multinomial Naive Bayes, que lida bem com a classificação de textos e não necessita de uma grande quantidade de dados para aprender os parâmetros. Além disso, ele possibilita que o modelo possa ser treinado rapidamente, quando comparado com modelos mais sofisticados.

Novas opiniões inseridas, passam por um processo de checagem para que se avalie se o texto é um ruído ou deve ter sua classificação predita, nessa etapa também é realizado o *Stemming* do texto, o qual visa reduzir cada palavra ao seu radical. Após esse processo, o texto então passa por uma limpeza onde são removidos links, caracteres especiais e palavras presentes em uma lista de *stopwords*. Em seguida, o texto é vetorizado, isto é, transformado em uma matriz a qual representa a frequência de cada um dos *tokens* do texto. Tal transformação é realizada a partir do vetor de probabilidade treinado com a base de treino do modelo.

Por fim, o vetor é passado para o modelo, o qual será responsável por prever a probabilidade de cada uma das classes, assim como a classe resultante da predição da frase. Após a predição as probabilidades resultantes de cada classe são, então, salvas no banco de dados e a classificação do texto pode ser inferida a partir da maior probabilidade gerada, esta última podendo variar entre 0 e 1.

Para a visualização dos organizadores é disponibilizado, na versão web, um "termômetro" que, com base no tipo de opiniões predominantes, indicará o estado do evento e servirá como um *feedback* rápido.

Apesar do processo de treinamento do modelo ser realizado de maneira sólida, o modelo ainda apresenta baixo rendimento em suas métricas de avaliação. Isso se deve ao fato da base de dados, referente ao domínio do MyMobiConf, não ser balanceada e não possuir a abundância necessária para contornar seu desbalanceamento. Dessa forma, o modelo acaba perdendo eficácia em suas predições.

### **4.3. Implementação da Transferência de Aprendizagem no Sistema MyMobiConf**

Antes de iniciar a implementação da transferência de aprendizagem, foi feita uma análise exploratória no *dataset* de opiniões tabuladas do MyMobiConf, o qual foi utilizado para o treinamento do último modelo implementado no sistema. Tais opiniões possuem um papel importante, pois serão responsáveis por ajudar a decidir quais dados, pertencentes à Ds, terão relevância para o treinamento do modelo.

O conjunto de dados totaliza 922 opiniões emitidas no sistema MyMobiConf e classificadas manualmente. Sendo dessas, 305 classificadas como positivas, 94 classificadas como negativas e 523 classificadas como neutras. Sendo assim, fica evidente o desbalanceamento dos dados utilizados para treinamento, o que conseqüentemente afeta as métricas de avaliação do modelo.

Para realizar a implementação da transferência de aprendizagem no modelo presente no MyMobiConf foi necessário, primeiro, obter um *dataset* com um grande número de registros e que já estivesse classificado entre positivo, negativo e neutro. Apesar da análise de sentimentos ser um tópico bastante trabalhado houve-se dificuldade em encontrar dados que estivessem em português e possuíssem as três classes, uma vez que a maior parte dos *datasets* possuíam uma classificação binária (positivo ou negativo). A falta de opções nesta etapa acarreta também a impossibilidade de obter-se dados com boa qua-

lidade para o modelo especificado, isto é, dados que melhor se assemelhem ao domínio alvo. O objetivo é que mesmo que o algoritmo TrAdaBoost realize a filtragem de dados relevantes, um conjunto de dados com maior semelhança resultará em um melhor resultado, uma vez que a quantidade de dados relevantes será maior e, por sua vez, a base de treino final terá, provavelmente, um vocabulário mais rico.

Foi utilizado então, um conjunto de dados de *tweets* disponibilizado na plataforma Kaggle, o qual possui um *dataset*<sup>2</sup> balanceado contendo 100 mil *tweets* já classificados entre positivos, negativos e neutros. O *dataset* em questão traz como informações o texto do *tweet*, sua data de publicação, a classificação de sentimento e um *emote* ou *hashtag* que os defina.

Após identificado o cenário como transferência de instâncias, para a utilização do algoritmo TrAdaBoost, foi utilizada a biblioteca ADAPT [de Mathelin et al. 2021], a qual implementa vários métodos de adaptação de domínio para cada uma das categorias citadas em 3.3. Dentre eles há a implementação do TrAdaBoost introduzida por [Dai et al. 2007]. O TrAdaBoost recebe como hiper-parâmetros um estimador que, caso não seja definido pelo usuário, será, por padrão, um regressor linear, os dados de entrada e saída do domínio alvo, o número de iterações a serem realizadas e a taxa de aprendizado que controlará a velocidade em que o peso das instâncias é atualizado.

Além disso, na preparação dos dados foi necessário também mudar a construção do vetorizador, já que, apesar do modelo desejado pertencer ao domínio de opiniões do MyMobiConf seu método *fit* fará uso dos dados do domínio fonte e, portanto, o vetorizador responsável por gerar os *tokens* deverá ser treinado com base nos dados do domínio fonte. É importante ressaltar que, os dados de entrada passados como hiper-parâmetros para o algoritmo também sofreram a vetorização com base no vetorizador gerado pelos registros do domínio fonte.

Para que fosse possível realizar uma análise do comportamento do algoritmo com base no número de iterações de treino, foi implementado uma validação cruzada estratificada, uma vez que a base de opiniões tabuladas do MyMobiConf deixa a desejar quanto a seu balanceamento. Tendo em vista que o modelo resultante atuará em  $D_t$  e os parâmetros passados para o método *fit* do modelo pertencem a  $D_s$ , foi necessário implementar uma função de validação cruzada customizada, haja vista que as implementações presentes nas bibliotecas tendem a realizar o método *fit* e *predict* com dados do mesmo domínio.

Assim sendo, as opiniões pertencentes ao conjunto de dados do MyMobiConf foram divididas, estratificadamente, na proporção 80%/20% em que 80% dos registros serão utilizados para treino (passados como hiper-parâmetro para o algoritmo) e os 20% restantes serão utilizados para teste. Destes 20% foi então utilizado a abordagem de *folds*, em que os dados foram divididos em 10 grupos, novamente estratificados por sua classe (negativo, positivo e neutro). Cada *fold*, portanto, recebeu 10% dos registros de cada classe presente no conjunto de teste. Com os dados de  $D_t$  já divididos, foi realizado a divisão dos dados de  $D_s$ , em que todo o conjunto de dados foi dividido, também, em 10 *folds* estratificadas por classes. Após a preparação dos dados, foi obtido como resultado os 10 conjuntos de dados de treino e teste que seriam, então, utilizados para a avaliação do modelo por meio da abordagem de validação cruzada.

---

<sup>2</sup><https://www.kaggle.com/datasets/augustop/portuguese-tweets-for-sentiment-analysis>

## 5. Resultados

Com os dados preparados e o algoritmo implementado, foi realizada uma análise exploratória dos resultados dos modelos treinados e validados com a validação cruzada. Para que fosse possível estudar o comportamento do algoritmo foi realizado uma análise empírica em relação aos resultados apresentados e o número de iterações de treino efetuadas.

Por meio do gráfico apresentado na Figura 1 é possível notar que a acurácia do modelo tende a aumentar até que se atinja um estado de *overfitting*, sendo este alcançado na configuração de 20 iterações. A partir deste momento os modelos gerados nas configurações superiores à 20 iterações passam a apresentar uma precisão cada vez menor em relação à anterior.

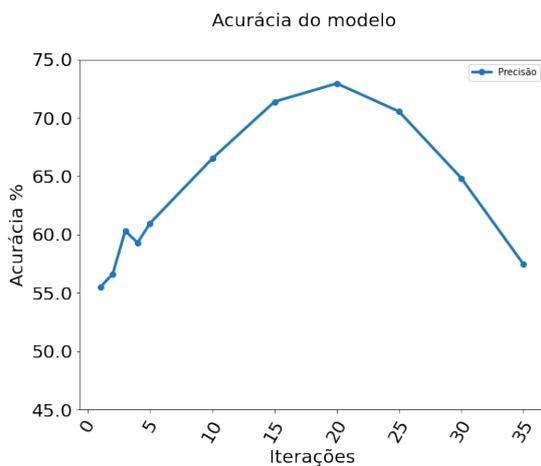
Analisando o gráfico de precisão por classe por número de iterações, encontrado em 2, nota-se que a precisão das classes positivas e neutras acaba não sofrendo uma variação tão significativa quando comparadas com a classe negativa. Isso se deve ao fato de que, inicialmente, o número de opiniões negativas era substancialmente inferior às demais classes. Por isso, à medida em que o número de iterações aumenta, devido a "filtragem" dos registros relevantes de Ds, mais rico se torna o vocabulário das opiniões no geral e devido à classe negativa possuir o vocabulário mais escasso, mais notável é a diferença de seu desempenho pelo aumento do número de iterações.

Quando avaliada a revocação do modelo na Figura 3, percebe-se uma queda brusca na porcentagem da revocação da classe negativa, comportamento já esperado tendo em vista que, com o aumento da precisão para a classe negativa, o número de registros classificados como "falso negativo" tende a diminuir e, como consequência, uma vez que a quantidade de falsos negativos é utilizada para calcular a revocação, a métrica de revocação também sofre uma diminuição. Haja vista que o aumento de uma das métricas leva, normalmente, à diminuição da outra, o modelo mostrou bons resultados no decorrer das iterações.

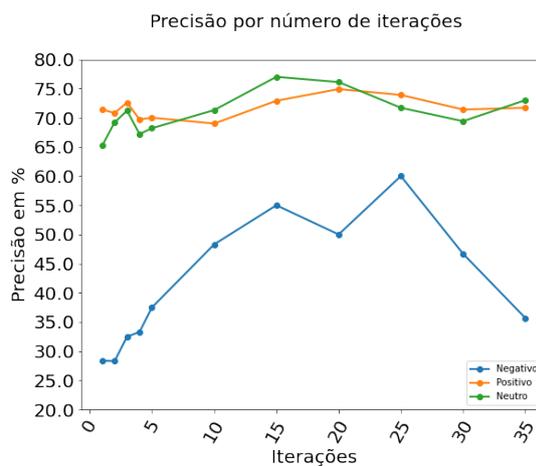
Observando o gráfico em 4, o qual apresenta a média harmônica entre revocação e precisão de cada uma das classes, percebe-se uma melhoria da métrica para as classes no decorrer do número de iterações.

Para um melhor comparativo de resultados, três cenários foram abordados, sendo o cenário 1: o treinamento e teste do modelo exclusivamente com a base do sistema MyMobiConf, o cenário 2: representado pelo treinamento do modelo com a base de *tweets* advindos do Kaggle e testado com a base de dados do MyMobiConf e, por último, o cenário 3: em que o modelo é gerado com base na implementação da transferência de aprendizagem por meio do TrAdaBoost.

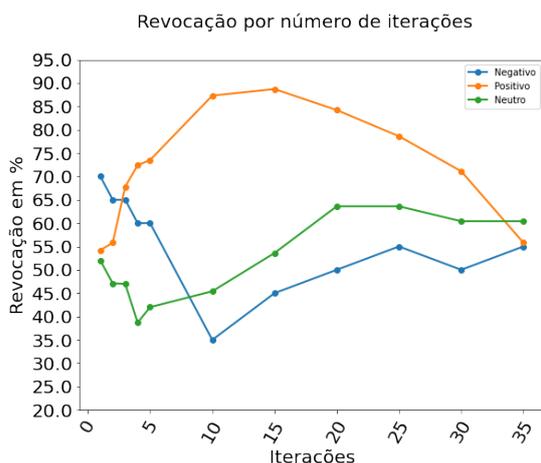
Observando os resultados apresentados pelo cenário 1, mostrados na tabela 1, percebe-se que o modelo gerado exclusivamente pela base de informações do MyMobiConf demonstra bons resultados classificando opiniões positivas e neutras, porém deixa a desejar em relação a classificação de opiniões negativas, alcançando uma precisão de apenas 20%, comportamento justificado pela escassez de opiniões negativas durante o treino. Levando em consideração o contexto ao qual o modelo se destina, uma boa precisão em relação às opiniões negativas é de extrema importância, tendo em vista que elas ajudarão os organizadores a identificar problemas e, conseqüentemente, promover um



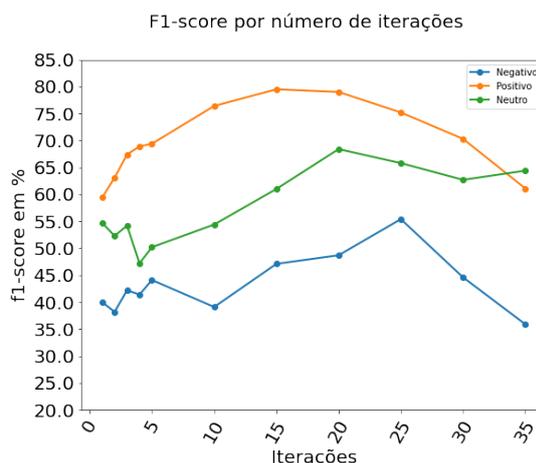
**Figura 1. Acurácia do modelo por número de iterações**



**Figura 2. Média de precisão de folds por número de iterações dividido por classe**



**Figura 3. Média de revocação de folds por número de iterações dividido por classe**



**Figura 4. Média de f1-score de folds por número de iterações dividido por classe**

melhor evento. Uma vez que a diminuição da precisão tende, normalmente, a aumentar a métrica de revocação, era esperado que a revocação para opiniões negativas fosse superior à sua precisão, porém seus resultados também deixam à desejar, alcançando apenas 2% de revocação, o que acaba justificando a baixa média harmônica (f1-score), que alcançou apenas 4%. Portanto, fica evidente que um modelo treinado exclusivamente com os dados do MyMobiConf, acaba não desempenhando muito bem, visto que no contexto inserido as opiniões negativas seriam as mais relevantes.

Levando em consideração um domínio genérico de opiniões, pode se ter a falsa impressão de que um aumento no número de opiniões, mesmo que não inseridas no domínio MyMobiConf, consequentemente resultará em uma melhora nas métricas de avaliação. Tal consideração é desconstruída pelos resultados apresentados pelo cenário 2, presentes na tabela 2, em que o modelo foi treinado com *tweets* não filtrados por tópico, e em seguida este modelo tentou avaliar opiniões do domínio MyMobiConf. Os resultados

	Precisão	Revocação	F1 Score
Positivo	58%	56%	57%
Negativo	20%	2%	4%
Neutro	66%	80%	72%
Macro-Average	48%	46%	44%

**Tabela 1. Métricas de Desempenho da Validação Cruzada para o Classificador Treinado Exclusivamente com a Base Disponibilizada pelo MyMobiConf**

	Precisão	Revocação	F1 Score
Positivo	48%	63%	55%
Negativo	10%	34%	16%
Neutro	59%	19%	28%
Macro-Average	39%	39%	33%

**Tabela 2. Métricas de Desempenho da Validação Cruzada para o Classificador Treinado Exclusivamente com a Base de Tweets disponibilizada pelo Kaggle e testada com opiniões tabuladas do MyMobiConf**

apresentados demonstram-se inferiores aos apresentados pelo cenário 1, uma vez que a precisão alcançada por todas as classes diminuiu e o modelo gerado não se mostra vantajoso, já que sua precisão faz com que seja mais interessante classificar a opinião jogando uma moeda à utilizar o modelo, haja vista que a acurácia alcançada pelo modelo foi inferior à 50%. Vale ressaltar também que o modelo, quando treinado e testado com a base de *tweets* do Kaggle, resultou em uma acurácia de 82,5%, o que mostra que a base de dados, em si, é uma boa base para treinamento.

Ao analisar os dois primeiros cenários, nota-se que o cenário 1 deixa a desejar na classificação de opiniões negativas e que o cenário 2, apesar de possuir uma base de treino mais ampla, acaba não gerando um bom resultado uma vez que as instâncias de treino e teste se encontram em domínios diferentes. Assim sendo, é apresentado o resultado do cenário 3, visualizado na tabela 3, o qual por meio do uso do algoritmo TrAdaBoost melhora a base de treino do modelo a partir das informações presentes na base de dados de *tweets*.

	Precisão	Revocação	F1 Score
Positivo	72%	64%	66%
Negativo	60%	55%	55%
Neutro	74%	79%	75%
Macro-Average	69%	66%	65%

**Tabela 3. Métricas de Desempenho da Validação Cruzada para o Classificador Treinado Utilizando TrAdaBoost**

A tabela 3 contém as métricas de avaliação do modelo gerado após 25 iterações do algoritmo, uma vez que esta configuração, além de retornar uma acurácia interessante, 70,5%, é também a que apresenta a maior precisão na classificação de opiniões negativas, como visualizado no gráfico 2, alcançando a marca de 60%. Analisando a tabela 3 e a comparando com o resultado dos demais cenários, observa-se que a implementação do

TrAdaBoost melhora consideravelmente as métricas de avaliação do modelo, destacando-se principalmente nas métricas que tangem opiniões negativas.

## 6. Conclusão e Trabalhos Futuros

O trabalho apresentado permitiu avaliar o comportamento de um modelo de análise de sentimentos e como este modelo comporta-se dependendo do domínio de suas instâncias de treino. Além disso, foi possível notar que a implementação da transferência de aprendizagem utilizando dados de um domínio externo pode trazer grandes ganhos para o modelo, desde que se avalie qual categoria de transferência de aprendizagem deve ser implementada e como o modelo comporta-se durante o processo. No algoritmo TrAdaBoost, em especial, deve-se atentar à como a variação do número de iterações afeta a evolução do modelo. Ademais, vale salientar que o uso indiscriminado da transferência de aprendizagem, sem que seja analisado o progresso do modelo, pode levar a um resultado inverso do esperado, chegando ao ponto de até mesmo deprevar seu desempenho, resultando em uma transferência negativa de aprendizagem.

Tendo em vista os aspectos observados e os resultados apresentados, a utilização de dados advindos de um domínio externo, podem sim ser benéficos para um modelo de análise de sentimentos, levando em consideração os ganhos das métricas apresentadas na tabela 3 e a comparação entre a acurácia do cenário 1, 63% , e do cenário 3, 70,5%.

A fim de continuar o trabalho apresentado visando melhorar as métricas do modelo destinado à classificação de opiniões de eventos, uma abordagem seria melhorar a qualidade dos dados utilizados no domínio fonte, construindo um *dataset* filtrado por comentários já relacionados à eventos, isso diminuiria a diferença entre os domínios e faria com que mais dados fossem considerados relevantes durante o processo de treinamento por meio do TrAdaBoost.

## Referências

- Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. (2007). Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, page 193–200, New York, NY, USA. Association for Computing Machinery.
- de Mathelin, A., Deheeger, F., Richard, G., Mougeot, M., and Vayatis, N. (2021). Adapt: Awesome domain adaptation python toolbox. *arXiv preprint arXiv:2107.03049*.
- Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.
- Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113.
- Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.