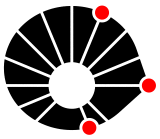

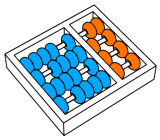

Análise Comparativa de Técnicas de Classificação de Textos de Spotted

Mateus Siqueira Batista - Jacques Wainer

Relatório Técnico - IC-PFG-24-16
Projeto Final de Graduação
2024 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Análise Comparativa de Técnicas de Classificação de Textos de Spotted

Mateus Siqueira Batista

Resumo

Este trabalho busca implementar e comparar diferentes técnicas de classificação de textos de 'spotted'. Spotteds são páginas de que publicam mensagens e textos de forma anônima em redes sociais. Para tal estudo, separou-se um conjunto de textos, rotulados entre 'postáveis' e 'não-postáveis'. A partir desses dados, testaram-se três abordagens distintas de classificação: utilizar o serviço de moderação pronto da OpenAI (Moderation Service), utilizar um modelo de linguagem pronto com um prompt específico, e utilizar embeddings em modelos de classificação tradicionais, como SVM, Naive Bayes e Random Forest. A métrica de recall foi utilizada para avaliar o desempenho inicial de cada abordagem, seguida de ajustes de hiperparâmetros para otimizar os resultados.

1 Introdução

Plataformas de "spotted"são comunidades de correio elegante digital comuns em universidades e presentes em redes sociais como Facebook e Instagram. Criadas para conectar pessoas da comunidade universitária, evoluíram para incluir mensagens variadas, desde flertes e desabafos até ameaças e discursos de ódio. Essas plataformas permitem que usuários postem mensagens anônimas sobre outros membros da comunidade, compartilhando experiências pessoais, opiniões ou relatos.

Com a popularização das páginas, tornou-se importante automatizar a análise e publicação dos textos nas redes sociais, garantindo uma maior disponibilidade do produto para os usuários. Nesse contexto, a moderação desses textos se torna cada vez mais crucial para assegurar um ambiente seguro e acolhedor. A presença de preconceitos, como racismo, homofobia, e outras formas de discriminação, bem como o uso de palavras de baixo calão e a exposição de pessoas de maneira inadequada, são fatores que inviabilizam a publicação desses textos em ambientes digitais públicos.

Sendo assim, a classificação automática de textos, levando em consideração esses aspectos semânticos, é uma ferramenta essencial para identificar e filtrar conteúdos inapropriados antes que sejam divulgados. A eficácia dessa classificação impacta diretamente a qualidade das interações online e a segurança dos usuários, prevenindo a propagação de discurso de ódio, assédio e informações que possam prejudicar indivíduos ou grupos.

Este trabalho busca explorar e comparar diferentes abordagens de classificação de textos de spotted, um tipo específico de postagem onde frequentemente se observa a necessidade de uma moderação rigorosa. A análise inclui a utilização de um serviço de moderação pronto

da OpenAI (Moderation Service), um modelo de linguagem com prompts específicos, e embeddings aplicados a modelos de classificação tradicionais, como SVM, Naive Bayes e Random Forest. A meta é avaliar a precisão e eficácia de cada método, buscando identificar a abordagem mais eficiente para a moderação automatizada de conteúdos em redes sociais.

2 Conceitos Relacionados

2.1 Spotteds

Plataformas de "spotted"são comunidades de mensagens anônimas amplamente difundidas no ambiente universitário. Encontradas nas principais redes sociais, como Facebook e Instagram, essas comunidades foram inicialmente criadas para facilitar a interação entre os membros da comunidade universitária. Contudo, com o tempo, passaram a abranger uma variedade maior de mensagens, que vão desde paqueras e desabafos até, lamentavelmente, ameaças, ofensas e discursos de ódio. Nessas plataformas, os usuários podem postar anonimamente sobre outros membros da comunidade, frequentemente compartilhando experiências pessoais, opiniões ou relatos.

O sistema construído pelo autor consiste em um software que permite o envio de "spotteds"e a sua publicação em redes sociais (Instagram e Facebook). Esse software também é responsável por classificar os textos enviados, garantindo que nada inadequado seja publicado, evitando que pessoas se sintam expostas de maneira desagradável ou ofendidas.

O envio dos "spotteds"ocorre em uma página HTML localizada no site oficial do Spotted (www.spotted.com.br). Os dados são salvos em um banco de dados, onde aguardam os processos de análise e classificação. Uma vez aprovados, os textos são publicados nas redes sociais por meio de APIs disponibilizadas, como a GraphAPI do Facebook e Instagram. Nosso estudo neste trabalho focará no processo de classificação dos textos.

2.2 Large Language Models (LLMs)

Large Language Models (LLMs) são avançados tipos de inteligência artificial projetados para processar e gerar texto como humanos. Baseados em redes neurais profundas, como o GPT da OpenAI, esses modelos são treinados em grandes conjuntos de dados textuais para entender e replicar padrões de linguagem. Eles são capazes de realizar uma variedade de tarefas, como tradução automática, resumo de textos, resposta a perguntas e criação de conteúdo original. A capacidade desses modelos de generalizar permite que eles abordem uma ampla gama de tópicos e se adaptem a diferentes contextos com pouca ou nenhuma modificação específica para a tarefa. No caso deste trabalho, utilizaremos um LLM junto a um prompt com contexto para processar o texto e identificar conteúdos sensíveis que não podem ser publicados em redes sociais de maneira pública.

A arquitetura em contextos de modelos de linguagem de LLMs refere-se à estrutura e organização interna desses sistemas computacionais. No caso específico dos LLMs como o GPT-4 e o BERT, a arquitetura é fundamental para determinar como esses modelos processam e geram texto. A arquitetura do Transformer, introduzida no artigo "Attention is All You Need"[1], por exemplo, é um paradigma chave utilizada por muitos LLMs modernos.

Ela é caracterizada por camadas de autoatenção que permitem que o modelo atente a diferentes partes do texto simultaneamente, capturando relações contextuais complexas. Essa arquitetura substituiu abordagens mais tradicionais baseadas em redes neurais recorrentes (RNNs) e convolucionais (CNNs), oferecendo uma capacidade significativamente maior de modelar dependências de longo alcance e relações não lineares entre as palavras. Portanto, a escolha e o refinamento da arquitetura são cruciais para o desempenho e a eficácia dos LLMs em tarefas como tradução automática, resumo de texto e geração de linguagem natural.

2.3 Embeddings

Em processamento de linguagem natural, embeddings referem-se à técnica de representação de palavras ou frases como vetores numéricos em um espaço contínuo. Esses vetores capturam semântica e relações contextuais entre palavras com base nos contextos em que aparecem nos dados de treinamento. Embeddings como o Word2Vec são amplamente utilizados para melhorar o desempenho de modelos de aprendizado de máquina em tarefas como classificação de texto, análise de sentimentos e tradução automática. Ao mapear palavras para vetores, os embeddings facilitam o processamento eficiente de linguagem natural e contribuem para uma melhor compreensão semântica das palavras em diferentes contextos linguísticos.

2.4 Algoritmos de Classificação e Aprendizado Supervisionado

Os algoritmos de classificação de machine learning por aprendizado supervisionado são uma classe de métodos utilizados para definir a categoria de uma nova amostra , com base em um conjunto de dados rotulados previamente. Esse tipo de aprendizado envolve a criação de um modelo que aprende a partir de um conjunto de dados de treinamento, onde as amostras são associadas a rótulos ou classes conhecidas. O objetivo é que o modelo, após ser treinado, consiga generalizar e fazer previsões precisas em novos dados não vistos anteriormente. Entre os algoritmos mais populares estão a Regressão Logistica, k-Nearest Neighbors (kNN), Naive Bayes, Suport Vector Machines (SVM) e Random Forest que vamos utilizar no nosso estudo para classificar os stpotteds.

A regressão logística é um algoritmo de aprendizado de máquina utilizado para problemas de classificação binária. Diferente da regressão linear, que prevê valores contínuos, a regressão logística prevê a probabilidade de uma observação pertencer a uma das duas classes possíveis. O modelo utiliza a função sigmoide para transformar a saída linear de uma combinação ponderada das características de entrada em um valor de probabilidade entre 0 e 1. Essa probabilidade é então usada para classificar a observação em uma das duas classes, geralmente aplicando um limiar de 0,5.

O k-Nearest Neighbors (kNN) é um algoritmo simples e intuitivo que classifica uma amostra com base nas classes das k amostras mais próximas no espaço de características. Esse método é eficaz em problemas onde a separação entre classes é complexa, pois não faz suposições sobre a distribuição dos dados. O Naive Bayes, por outro lado, baseia-se no teorema de Bayes e na suposição de independência condicional entre as características. Apesar dessa suposição ser frequentemente irrealista na prática, o Naive Bayes tem bom

desempenho em muitos problemas reais, especialmente em classificação de textos e filtragem de spam.

As Suport Vector Machines (SVM) buscam encontrar o hiperplano ótimo que separa as classes de forma maximamente distante, aumentando assim a margem de separação entre elas. Este método é eficaz em problemas de alta dimensionalidade e pode ser ajustado para classificação não linear utilizando truques como o kernel trick. O Random Forest é um método de ensemble que combina múltiplas árvores de decisão para melhorar a precisão e a robustez das previsões. Ele reduz a variância e evita overfitting, resultando em modelos mais poderosos e generalizáveis. Esses algoritmos, quando aplicados corretamente, podem transformar grandes volumes de dados em insights valiosos, auxiliando na tomada de decisões em diversas áreas, como medicina, finanças e marketing.

No nosso trabalho, vamos utilizar os spotteds já classificados manualmente como dados de treinamento do modelo. Esses dados estão rotulados em duas categorias distintas: "postáveis"e "não-postáveis", ou "bons"e "ruins". Essa classificação prévia permite que o algoritmo aprenda a reconhecer padrões e características que distinguem os spotteds de cada categoria, aprimorando a capacidade do modelo em fazer previsões precisas em novos dados não rotulados.

3 Metodologia

Para investigar a eficácia de diferentes abordagens de classificação de textos de spotteds, foi feito um experimento estruturado em três fases principais: preparação dos dados, aplicação dos métodos de classificação e avaliação dos resultados.

3.1 Preparação dos dados

Inicialmente, foram selecionados manualmente um conjunto de 1600 textos provenientes dos bancos de dados da plataforma spotted, divididos igualmente entre duas categorias:

- **Textos bons:** Mensagens apropriadas para publicação, sem a presençã de conteúdos proibídos.
- **Textos ruins:** Mensagens contendo conteúdos proibídos.

Assim, construimos um outro branco de dados com 800 textos bons e 800 ruins. Os conteúdos considerados proibídos são:

- Ofensivo (bullying ou ataque a alguma pessoa ou grupo).
- Ameaças (qualquer tipo de ameaça dirigida a alguém, locais públicos, organizações, etc.).
- Deprimido (sinais de extrema tristeza com intenções suicidas).
- Publicidade (comprar, vender ou oferecer qualquer tipo de serviço, pedir seguidores, promover alguém ou um evento, links, urls, hashtags).

- Exposição (qualquer tipo de difamação de alguém, por fazer algo não ético ou repugnante à sociedade, trapaceiro, desonesto, não leal, injusto).
- Política (textos envolvendo políticos famosos, decisões polêmicas de esquerda ou direita, etc).
- Constrangedor (algo que pode expor alguém negativamente).
- Discurso de ódio (ataque direto a grupos minoritários ou apoio a organizações opressivas).
- Convite para relações sexuais ou algo explícito.
- Alguém que procura um objeto perdido ou procura o dono de algum objeto encontrado.

Esses textos foram cuidadosamente classificados por moderadores humanos experientes, garantindo a precisão das categorias. Essa base de dados balanceada serve como referência para testar e comparar os métodos de classificação e tem exemplos de todas as categorias mencionadas acima.

3.2 Aplicação dos Métodos de Classificação

Nesta etapa, submetemos os textos preparados aos métodos de classificação. Nesse caso, três abordagens distintas foram implementadas:

3.2.1 Serviço de Moderação da OpenAI (Moderation Service)

Utilizamos o serviço pronto de moderação da OpenAI, que analisa o conteúdo textual e identifica a presença de elementos impróprios. Cada texto da base de dados foi submetido ao serviço, e as classificações retornadas foram registradas.

O Moderation Service da OpenAI é uma ferramenta para verificar se um texto é potencialmente prejudicial. Desenvolvedores podem usá-la para identificar e filtrar conteúdos nocivos. O modelo classifica as seguintes categorias:

- Ódio: Conteúdo que expressa, incita ou promove ódio baseado em raça, gênero, etnia, religião, nacionalidade, orientação sexual, status de deficiência ou casta. Ódio direcionado a grupos não protegidos é considerado assédio.
- Ódio/Ameaçador: Conteúdo de ódio que inclui violência ou sério dano ao grupo alvo baseado em raça, gênero, etnia, religião, nacionalidade, orientação sexual, status de deficiência ou casta.
- Assédio: Conteúdo que expressa, incita ou promove linguagem de assédio contra qualquer alvo.
- Assédio/Ameaçador: Conteúdo de assédio que inclui violência ou sério dano ao alvo.

- Autoagressão: Conteúdo que promove, encoraja ou descreve atos de autoagressão, como suicídio, cortes e distúrbios alimentares.
- Autoagressão/Intenção: Conteúdo onde o autor expressa que está envolvido ou pretende se envolver em atos de autoagressão.
- Autoagressão/Instruções: Conteúdo que encoraja a realizar atos de autoagressão ou dá instruções sobre como cometer tais atos.
- Sexual: Conteúdo destinado a excitar sexualmente, como a descrição de atividades sexuais, ou que promove serviços sexuais (excluindo educação e bem-estar sexual).
- Sexual/menores: Conteúdo sexual que inclui indivíduos com menos de 18 anos.
- Violência: Conteúdo que descreve morte, violência ou lesões físicas.

O endpoint do Moderation Service é gratuito e foi acessado via API por meio da biblioteca da OpenAI feita para o python.

3.3 Modelo de Linguagem com Prompt

Aplicamos um modelo de linguagem (GPT-3.5 Turbo), utilizando um prompt específico para direcionar a análise do conteúdo. O prompt foi configurado para auxiliar o modelo a responder se um texto é apropriado ou não, considerando os conteúdos proibídos listados anteriormente. Os resultados foram coletados para análise comparativa.

3.4 Embeddings e Modelos de Classificação Tradicionais

Para esta abordagem, convertemos os textos em vetores de embeddings utilizando a biblioteca *sentence-transformers/all-MiniLM-L6-v2* em Python. Esses embeddings foram divididos entre dados de treinamento e teste. Os dados de treinamento foram usados como entradas para treinar os algoritmos de classificação tradicionais, implementados pela biblioteca *scikit-learn*. Os dados teste foram utilizados para calcular as métricas e o desempenho de cada algoritmo. Os algoritmos de classificação utilizados foram Regressão Logistica, kNN, Random Forest, Naive Bayes e SVM.

3.5 Avaliação dos Resultados

Para comparar a eficácia das abordagens, analisamos as métricas de desempenho de cada método. As principais métricas consideradas incluem:

- Acurácia: A porcentagem de classificações corretas.
- Precisão: A proporção de textos classificados como ruins que realmente são ruins.
- Recall: A proporção de textos ruins identificados corretamente.
- F1-Score: A média harmônica da precisão e recall, fornecendo uma única métrica de desempenho.

No nosso caso, como a aprovação e porterior publicação de um texto ruim deve ser evitada ao máximo, a métrica do **recall** foi levada mais em conta na análise.

Por fim, para potencializar os resultados, foi feito um *grid-serarch* com python. Com isso, para cada algoritmo, os melhores hiperparâmetros foram encontrados.

4 Resultados

Para os resultados, criamos matrizes de confusão e calculamos as metricas mencionadas anteriormente.

- Resultado Moderation Service

  A matriz de confusão da Fig. 1 mostra a distribuição dos resultados.

  - Acurácia: 0.578,
  - Precisão: 0.943,
  - Recall: 0.167,
  - F1-score: 0.283

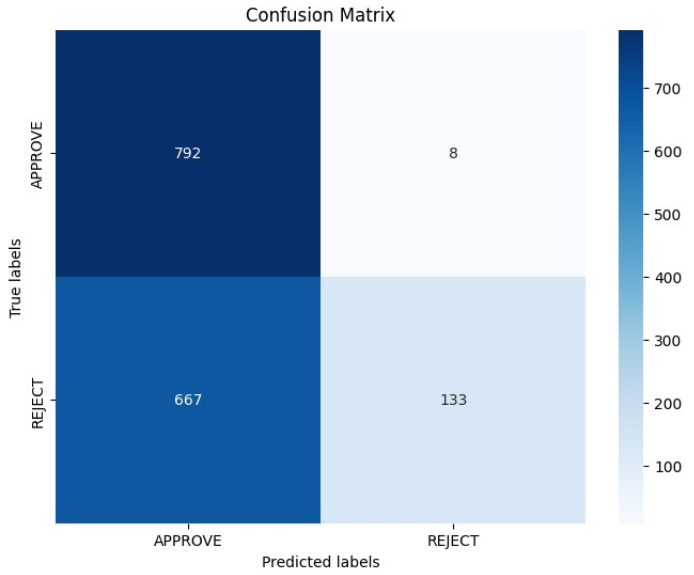

Figura 1: Matriz de confusão dos resultados para o Moderation Service da OpenAI

- Resultado GPT-3.5 Turbo

  A matriz de confusão da Fig. 2 mostra a distribuição dos resultados.

  - Acurácia: 0.701,
  - Precisão: 0.711,

- Recall: 0.679,
- F1-score: 0.694

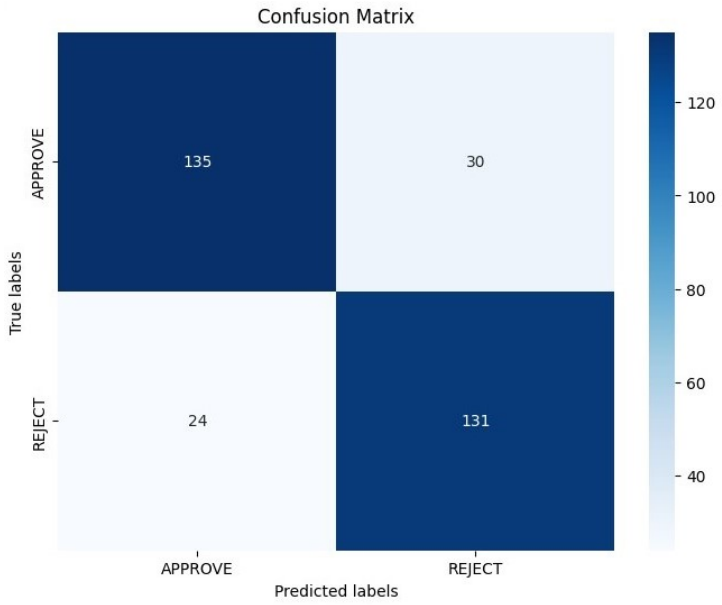

Figura 2: Matriz de confusão dos resultados para o GPT-3.5 Turbo com prompt

- Resultado Regressão Logística

  A matriz de confusão da Fig. 3 mostra a distribuição dos resultados.

  - Acurácia: 0.837,
  - Precisão: 0.828,
  - Recall: 0.838,
  - F1-score: 0.833

- Resultado Random Forest

  A matriz de confusão da Fig. 4 mostra a distribuição dos resultados.

  - Acurácia: 0.837,
  - Precisão: 0.828,
  - Recall: 0.838,
  - F1-score: 0.833

- Resultado Naive Bayes

  A matriz de confusão da Fig. 5 mostra a distribuição dos resultados.

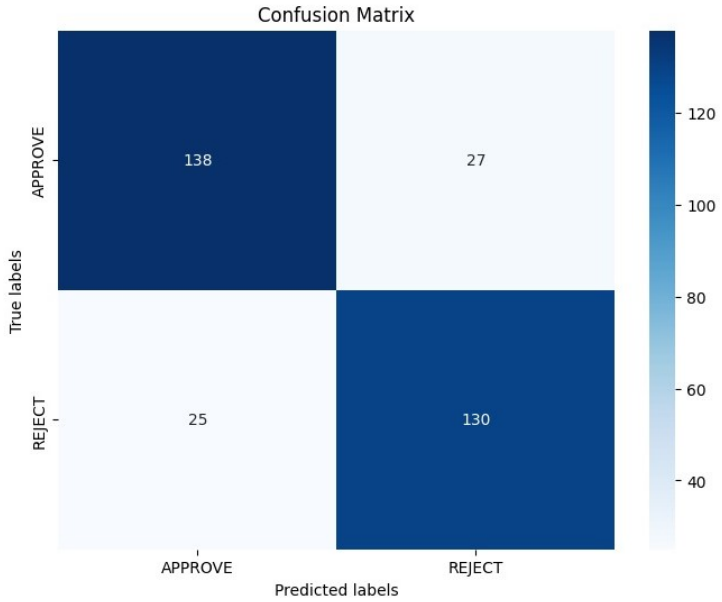

Figura 3: Matriz de confusão dos resultados para a Regressão Logística

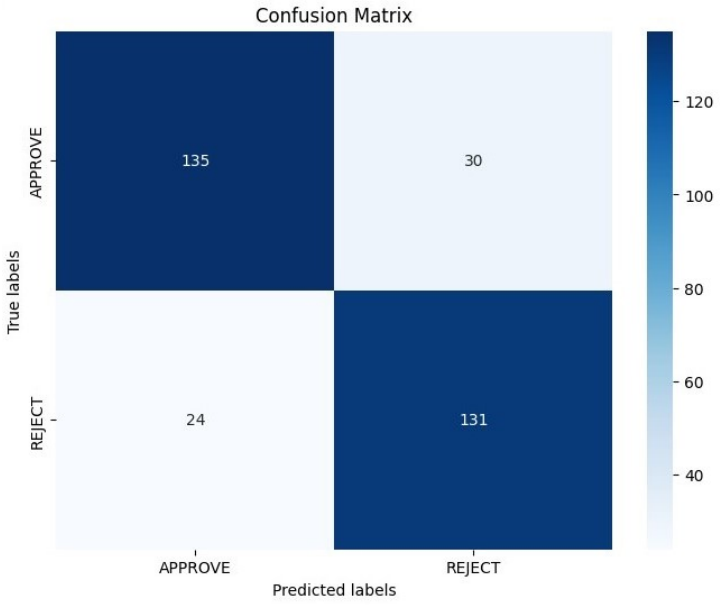

Figura 4: Matriz de confusão dos resultados para o Random Forest

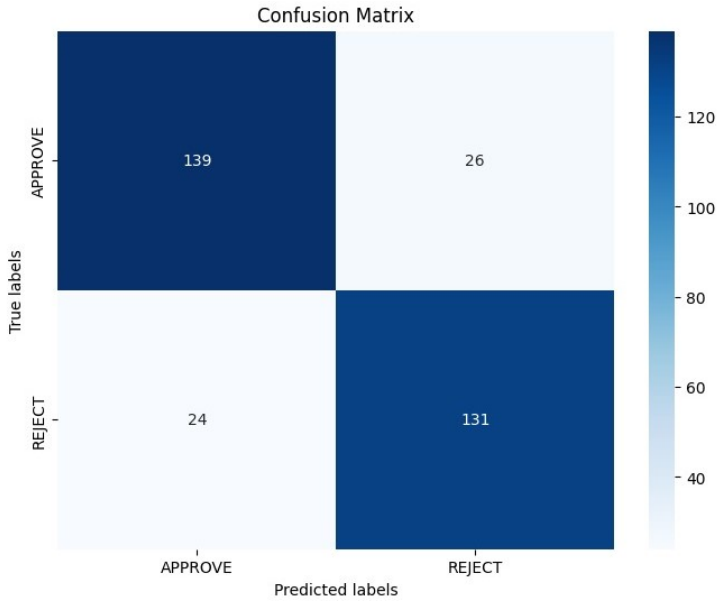

Figura 5: Matriz de confusão dos resultados para o Naive Bayes

- Acurácia: 0.843,
- Precisão: 0.834,
- Recall: 0.845,
- F1-score: 0.84

- Resultado SVM

  A matriz de confusão da Fig. 6 mostra a distribuição dos resultados.

  - Acurácia: 0.85,
  - Precisão: 0.836,
  - Recall: 0.858,
  - F1-score: 0.8471

- Resultado kNN

  A matriz de confusão da Fig. 7 mostra a distribuição dos resultados.

  - Acurácia: 0.806,
  - Precisão: 0.878,
  - Recall: 0.697,
  - F1-score: 0.777

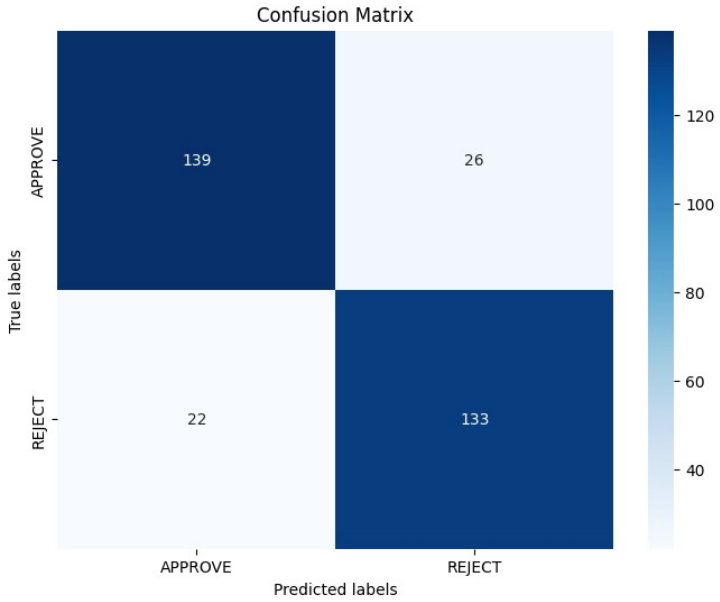

Figura 6: Matriz de confusão dos resultados para o SVM

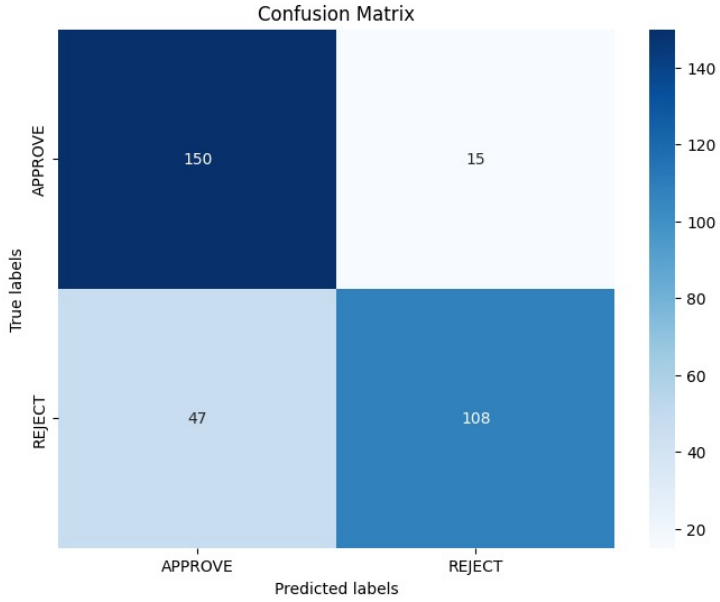

Figura 7: Matriz de confusão dos resultados para o kNN

Em uma primeira análise, os algoritmos de classificação tradicionais com embeddings tiveram uma performace significativamente superior, epecialmente o Naive Bayes e o SVM, com recalls respectivamente iguais a 0.845 e 0.858. O Moderation Service da OpenAI não alcançou bons resultados pois sua análise não classifica questões como publicidade ou política, assuntos que são considerados poibídos.

Analisando a fundo os textos classificados incorretamente pelo GPT, foi constatado que o modelo tem dificuldades de identificar conteúdos de publicidade, que muitas vezes são enviados de maneira intencional se passando por uma pergunta sobre um evento, produto ou serviço. O spotted "Vocês ficaram sabendo que a festa x vai ter a bebida y?"é um exemplo disso, pois divulga uma festa na forma de dúvida dirigida ao público, enganando a classificação do modelo.

Então, para potencializar os resultados dos algoritmos tradicionais, foi utilizada a técnica do *grid-search* para buscar os seus melhores hiperparâmetros. Para o classificador Naive Bayes, foram explorados diferentes valores para o parâmetro *priors*, sendo a configuração *[0.1, 0.9]* com melhor desempenho, alcançando um recall de aproximadamente 0.9. Em seguida, foi aplicado o mesmo processo para o SVC, agora buscando os melhores valores para os parâmetros de regularização C e *gamma*. Após a busca, os valores ideais foram *C=1* e *gamma=0.03125*, resultando em um recall de 0.89.

5 Conclusão

Neste projeto, foi realizada uma análise comparativa de diferentes técnicas de classificação de textos de spotted. Foram testadas três abordagens principais: o serviço de moderação da OpenAI, o GPT-3.5 Turbo e algoritmos de classificação tradicionais utilizando embeddings. Os resultados mostraram que os algoritmos de classificação tradicionais, especialmente o Naive Bayes e o SVM, apresentaram desempenho superior, com recalls de 0.845 e 0.858, respectivamente .

O serviço de moderação da OpenAI apresentou desempenho insatisfatório, com um recall de apenas 0.167, principalmente devido à sua incapacidade de identificar conteúdos relacionados a publicidade e política, que são considerados proibidos . Por outro lado, o GPT-3.5 Turbo teve um desempenho razoável, com recall de 0.679 e F1-score de 0.694, mas mostrou dificuldades em identificar conteúdos publicitários disfarçados de perguntas.

Para melhorar o desempenho do GPT, uma das abordagens sugeridas é aumentar a quantidade de exemplos de spotteds utilizados para contextualizar a tarefa. A inclusão de mais exemplos pode ajudar o modelo a aprender melhor as nuances dos textos e melhorar sua capacidade de classificação.

Além disso, uma abordagem promissora seria a utilização de um sistema híbrido, combinando múltiplos filtros de classificação. Nesse sistema, o serviço de moderação da OpenAI poderia ser utilizado para identificar conteúdos explicitamente prejudiciais, como assédio, autoagressão e violência. O GPT, por sua vez, seria responsável pela classificação do restante dos textos. Dessa forma, o sistema de moderação da OpenAI agiria como um filtro preliminar, removendo conteúdos altamente nocivos, enquanto o GPT realizaria uma análise mais refinada e detalhada dos textos restantes.

Essa abordagem pode maximizar a eficácia da moderação, aproveitando os pontos fortes de cada tecnologia e mitigando suas fraquezas. Além disso, o uso combinado de diferentes classificadores pode aumentar a precisão e a robustez do sistema de classificação de textos de spotted.

Referências

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, *Attention Is All You Need* (Jun 2017) .

[2] Christopher M. Bishop. "Pattern Recognition and Machine Learning."Springer, 2006.

[3] Jiawei Han, Micheline Kamber, Jian Pei. "Data Mining: Concepts and Techniques."Morgan Kaufmann, 2011.

[4] Tom Mitchell. "Machine Learning."McGraw-Hill, 1997.

[5] T. Joachims. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features."Proceedings of the 10th European Conference on Machine Learning (ECML-98), Springer-Verlag, 1998.