Trabalho de Conclusão de Curso

Injeção de Ruído em Serviços de Aprendizagem de Máquina na Nuvem

Lucas de Oliveira Amorim
loa@ic.ufal.br

Orientador:

Prof. Dr. Baldoino Fonseca dos Santos Neto

Maceió, Julho de 2023

Lucas de Oliveira Amorim

Injeção de Ruído em Serviços de Aprendizagem de Máquina na Nuvem

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação do Instituto de Computação da Universidade Federal de Alagoas.

Orientador:

Prof. Dr. Baldoino Fonseca dos Santos Neto

Maceió, Julho de 2023

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecário: Jorge Raimundo da Silva – CRB-4 –1320

A524i Amorim, Lucas de Oliveira.
Injeção de Ruído em Serviços de Aprendizagem de Máquina na Nuvem / Lucas de Oliveira Amorim. – 2023.
46 f.

Orientador: Baldoino Fonseca dos Santos Neto.
Monografia (Trabalho de Conclusão de Curso em Ciência da Computação) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2023.

Bibliografia: f. 42-46.

1. Machine Learning as a Service (MLaaS). 2. Análise de sentimentos. 3. Ruído em dados de texto. 4. Robustez de modelos. I. Título.

CDU: 519.683.5

Agradecimentos

Quero expressar minha mais sincera gratidão à minha mãe, Lúcia Amorim, e ao meu pai, Cícero Amorim. Sua inabalável fé em minha capacidade, seu amor incondicional e seu apoio contínuo foram a minha força motriz durante esta jornada. A eles, devo minha eterna gratidão.

Aos meus irmãos, que sempre estiveram ao meu lado e me deram suporte em todos os momentos, expresso minha profunda admiração e agradecimento. A camaradagem, a compreensão e o amor que vocês demonstraram foram essenciais para a minha trajetória.

Gostaria de expressar meu sincero agradecimento ao Professor Baldoino. Sua excelência como educador e seu profundo conhecimento foram essenciais para minha formação acadêmica. Seus insights valiosos e conselhos perspicazes foram catalisadores na melhoria da qualidade desta pesquisa.

Um agradecimento especial vai para Juliano Rocha, cuja colaboração foi crucial para a realização deste trabalho. Sua disposição em ajudar e sua expertise foram de imensa ajuda, e sou verdadeiramente grato por sua contribuição.

Aos meus amigos próximos, Nelson Gomes, Gabriel Barbosa, Daniel Vassalo, Lucas Raggi, Wagner Fontes e França MacDowell, agradeço de coração. Seu apoio contínuo, amizade e momentos de descontração foram um bálsamo durante os momentos mais estressantes desta jornada. Vocês enriqueceram minha experiência e contribuíram para o meu crescimento pessoal e acadêmico.

Em resumo, a realização deste trabalho não seria possível sem a ajuda e o apoio de todas as pessoas maravilhosas mencionadas acima. Sua generosidade, em termos de tempo, expertise e apoio emocional, fez toda a diferença para mim. Sou verdadeiramente agradecido.

Resumo

A aplicação de Machine Learning as a Service (MLaaS) para a análise de sentimentos é um elemento crucial no panorama tecnológico atual, impactando diversas áreas, desde o marketing digital até a pesquisa acadêmica. Este estudo concentrou-se na exploração de como a injeção de ruído em dados de texto afeta o desempenho desses serviços, utilizando como exemplos os serviços oferecidos por três grandes provedores: Microsoft, Amazon e Google.

O objetivo principal desta pesquisa foi avaliar a robustez e confiabilidade dessas plataformas frente a vários tipos de ruídos induzidos através da biblioteca NlpAug. As métricas de desempenho, em particular a F-measure, foram analisadas detalhadamente e os resultados obtidos foram comparados entre os diferentes serviços.

Os resultados obtidos demonstraram uma queda linear na F-measure à medida que o nível de ruído nos dados aumentava. Embora a Amazon e a Google tenham apresentado um desempenho ligeiramente superior em relação à Microsoft, todas sofreram uma redução significativa de performance à medida que os níveis de ruído aumentavam.

Este estudo evidencia a importância de estratégias robustas de pré-processamento e limpeza de dados ao implementar soluções de machine learning. Além disso, foi observado que certos tipos de ruído, como erros de digitação, OCR e substituição aleatória de caracteres, causam uma piora da performance dos modelos de análise de sentimentos comparado a outros.

Concluindo, esta pesquisa fornece insights fundamentais sobre a necessidade de um entendimento mais aprofundado das capacidades e limitações dos serviços de MLaaS, a fim de utilizá-los de forma mais efetiva em aplicações reais. Além disso, abre caminho para futuras investigações que podem explorar a robustez desses serviços em relação a outros tipos de dados e tarefas de machine learning.

Palavras-chave: Machine Learning as a Service (MLaaS), Análise de sentimentos, Ruído em dados de texto, Robustez de modelos.

Abstract

The application of Machine Learning as a Service (MLaaS) for sentiment analysis is a critical element in today's technological landscape, impacting various areas from digital marketing to academic research. This study focused on exploring how the injection of noise into text data affects the performance of these services, using as examples the services offered by three major providers: Microsoft, Amazon, and Google.

The primary goal of this research was to evaluate the robustness and reliability of these platforms against various types of noise induced through the NlpAug library. Performance metrics, particularly the F-measure, were analyzed, and the results obtained were compared among the different services.

The results demonstrated a linear drop in the F-measure as the noise level in the data increased. Although Amazon and Google showed slightly superior performance compared to Microsoft, all suffered a significant performance decrease as noise levels increased.

This study highlights the importance of robust pre-processing and data cleaning strategies when implementing machine learning solutions. Moreover, it was observed that certain types of noise, such as typographical errors, OCR, and random character substitution, worsens the performance of sentiment analysis models compared to others.

In conclusion, this research provides essential insights into the need for a deeper understanding of the capabilities and limitations of MLaaS services in order to use them more effectively in real applications. Furthermore, it paves the way for future investigations that may explore the robustness of these services in relation to other types of data and machine learning tasks.

Key-words: Machine Learning as a Service (MLaaS), Sentiment analysis, Text Noise Injection, Model reliability.

Lista de Figuras

Lista de Tabelas

Conteúdo

1

Introdução

A aprendizagem de máquina e a inteligência artificial revolucionaram a maneira como os dados são processados e entendidos, permitindo uma nova era de capacidades preditivas e analíticas (Goodfellow et al., 2016). No entanto, à medida que essas tecnologias continuam a evoluir e a serem integradas em uma miríade de aplicações, a robustez, a confiabilidade e a precisão dos modelos de aprendizado de máquina se tornam de importância crítica (Chen, 2021).

A natureza distribuída da computação na nuvem permitiu um rápido crescimento na oferta de serviços de aprendizagem de máquina como um serviço (MLaaS). Empresas como Microsoft, Amazon e Google oferecem agora poderosos serviços MLaaS que podem ser facilmente integrados em qualquer aplicação, sem a necessidade de competências especializadas em aprendizagem de máquina ou computação de alto desempenho (Ribeiro et al., 2015). Estes serviços incluem plataformas como o Azure Cognitive Services da Microsoft (Microsoft, 2023), o Amazon Comprehend (Amazon, 2023), e o Google Cloud Natural Language API (Google, 2023), todos os quais oferecem análise de sentimento, entre outros recursos de processamento de linguagem natural.

A análise de sentimento é uma área em rápida evolução do processamento de linguagem natural que visa identificar e extrair opiniões subjetivas de textos (Liu, 2012). Os serviços de análise de sentimento são frequentemente utilizados para, por exemplo, entender opiniões públicas, avaliar satisfação de clientes, monitorar a marca e produtos, entre outros.

Neste trabalho, exploraremos a injeção de ruídos em datasets utilizando a biblioteca NlpAug (Ma, 2021). A NlpAug é uma biblioteca de aumento de texto para aprendizado de máquina na linguagem natural, que ajuda a gerar novos dados de treinamento ao introduzir vários tipos de ruídos, como troca de palavras, substituição de sinônimos, remoção de palavras e mais (Ma, 2021). A introdução de ruídos nos dados é uma forma eficaz de aumentar a robustez dos modelos de aprendizado de máquina, testando a capacidade dos modelos de lidar com dados de entrada imprecisos ou com ruído (Sáez et al., 2016).

1.1 Justificativa

A importância da avaliação e compreensão dos limites e capacidades dos serviços de Machine Learning como um Serviço (MLaaS) não pode ser subestimada. Com a crescente dependência desses serviços em uma ampla variedade de aplicações do mundo real, é vital entender como esses serviços respondem a variações inesperadas de dados ou ruídos introduzidos.

A era da informação e a abundância de dados de texto disponíveis, especialmente nas redes sociais, aumentaram a necessidade de serviços de análise de sentimento robustos e confiáveis. No entanto, a presença de ruído e outras inconsistências nesses dados pode impactar significativamente a performance desses serviços.

Adicionalmente, o campo de aprendizado de máquina tem experimentado um crescimento no número e na sofisticação dos ataques adversariais. Esses ataques exploram vulnerabilidades nos modelos de aprendizado de máquina para induzi-los a tomar decisões errôneas, o que pode ter consequências devastadoras em cenários do mundo real (Kurakin et al., 2016).

Por último, a injeção de ruídos nos dados pode ajudar a melhorar a robustez e confiabilidade dos modelos de aprendizado de máquina, testando a capacidade dos modelos de lidar com entradas imprecisas ou com ruído. Esta abordagem tem sido aplicada em muitos domínios da aprendizagem de máquina e tem mostrado ser um método eficaz para melhorar o desempenho do modelo (Sheng et al., 2008).

1.2 Objetivos

1.2.1 Objetivo Geral

O principal objetivo deste trabalho é explorar o impacto da injeção de ruídos em dados de texto no desempenho dos serviços de Machine Learning as a Service (MLaaS) na análise de sentimento. Isso será realizado por meio do uso de um conjunto de dados de tweets e da biblioteca NlpAug para introduzir vários tipos de ruído. A avaliação da robustez e da confiabilidade dos serviços MLaaS, em resposta a essas injeções de ruídos, ajudará a entender melhor suas limitações e potencialidades.

1.2.2 Objetivos Específicos

1. Avaliar a resposta dos serviços MLaaS da Microsoft, Amazon e Google para a análise de sentimento em relação a diferentes tipos de ruído introduzidos.

2. Comparar a robustez e confiabilidade dos serviços MLaaS entre si e destacar suas respectivas forças e fraquezas na análise de sentimentos.

3. Determinar se diferentes tipos de ruído têm impactos distintos nos resultados dos serviços

de análise de sentimento e, em caso afirmativo, quais tipos de ruído são mais prejudiciais ao desempenho dos serviços MLaaS.

2

Fundamentação teórica

Este capítulo explora em detalhes os conceitos e tecnologias fundamentais para a compreensão do nosso estudo. Inicialmente, discutimos os conceitos de aprendizagem de máquina e processamento de linguagem natural, em seguida, os serviços de Machine Learning como um Serviço (MLaaS), com foco particular nos serviços de análise de sentimentos.

Na sequência, abordamos o conceito de ruído em conjuntos de dados e como ele pode impactar a performance dos serviços de MLaaS. Em seguida, apresentamos alguns dos tipos de ruído que podem ser adicionados a um texto.

2.1 Aprendizado de Máquina (Machine Learning)

O Aprendizado de Máquina (Machine Learning), um subcampo da Inteligência Artificial, é um método de análise de dados que automatiza o processo de construção de modelos analíticos. Ele é baseado na ideia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana (Samuel, 1959).

Em um sistema de aprendizado de máquina, um algoritmo é treinado em um conjunto de dados ou corpus. Durante este processo, o algoritmo aprende a partir dos exemplos de entrada e respectivas saídas esperadas, adaptando-se progressivamente para melhor desempenho. Após o treinamento, o sistema pode aplicar o conhecimento aprendido a novos dados para prever resultados, categorizar entradas ou descobrir estruturas subjacentes, dependendo do tipo de aprendizado e da tarefa específica (Mitchell, 1997).

Existem três tipos principais de aprendizado de máquina: supervisionado, não supervisionado e reforço. No aprendizado supervisionado, o algoritmo é treinado em um conjunto de dados que já contém a resposta correta, conhecida como rótulo (label). No aprendizado não supervisionado, o algoritmo é usado para explorar dados que não foram rotulados, classificados ou categorizados, e o sistema precisa trabalhar por conta própria para descobrir a estrutura subjacente nos dados. O aprendizado por reforço é um tipo de aprendizado de máquina onde um

agente aprende a tomar decisões ao receber recompensas ou punições por suas ações (Mitchell, 1997).

Os algoritmos de aprendizado de máquina têm uma ampla variedade de aplicações, incluindo motores de busca, detecção de fraudes, recomendações de produtos, análise de sentimentos e diagnósticos médicos, entre outros (Jordan and Mitchell, 2015).

2.2 Processamento de Linguagem Natural (NLP)

O Processamento de Linguagem Natural (NLP, do inglês Natural Language Processing) é um subcampo da ciência da computação e inteligência artificial que se concentra na interação entre computadores e linguagem humana. Em particular, o NLP se preocupa com a programação de computadores para processar e analisar grandes quantidades de dados de linguagem natural (Jurafsky and Martin, 2009).

O objetivo do NLP é permitir que os computadores entendam, interpretem e manipulem a linguagem humana. Isso inclui várias tarefas, como tradução automática (traduzir texto de um idioma para outro), reconhecimento de fala (converter fala em texto), geração de texto (produzir texto que faça sentido a partir de dados), extração de informações (obter informações estruturadas a partir de texto não estruturado), e análise de sentimentos (determinar o sentimento expresso em um certo texto) (Liu, 2012).

O NLP é uma área de pesquisa multidisciplinar que faz uso de técnicas de linguística, ciência da computação e inteligência artificial. A implementação de técnicas de NLP em aplicações do mundo real é complexa devido à natureza ambígua e variável da linguagem humana. No entanto, graças aos avanços recentes em aprendizado de máquina e ao aumento da disponibilidade de grandes corpora de texto, a eficácia e a aplicabilidade das técnicas de NLP têm crescido significativamente (Chaudhary et al., 2023).

2.3 Machine Learning as a Service (MLaaS)

Nos últimos anos, o conceito de Machine Learning as a Service (MLaaS) tem ganhado muita atenção. As plataformas de MLaaS proporcionam uma forma eficiente e escalável de aplicar técnicas de aprendizado de máquina, sem a necessidade de conhecimento especializado em desenvolvimento e manutenção de algoritmos de machine learning (Hashem et al., 2015).

Os provedores de MLaaS oferecem serviços para várias tarefas, como reconhecimento de voz, visão por computador, análise de sentimento, tradução automática, entre outros. Estes serviços, embora variem em termos de funcionalidade e performance, oferecem uma forma conveniente e eficiente para empresas e indivíduos aplicarem o poder da aprendizagem de máquina em suas operações sem a necessidade de grandes investimentos em hardware ou especialistas em aprendizado de máquina.

2.3.1 Serviços de Análise de Sentimento

Entre os vários serviços de MLaaS, a análise de sentimento é um dos mais utilizados, devido à sua aplicabilidade em uma ampla gama de campos, incluindo marketing, atendimento ao cliente, mídias sociais, entre outros (Liu, 2012).

A análise de sentimento, também conhecida como mineração de opinião, refere-se ao uso de processamento de linguagem natural (NLP), análise de texto e técnicas de aprendizado de máquina para identificar e extrair informações subjetivas de fontes de texto. Ela permite identificar a atitude, opinião ou emoção expressa em um texto, seja ele uma frase, um comentário, uma revisão de produto ou mesmo um tweet (Liu, 2012).

A Microsoft, a Amazon e o Google, por exemplo, oferecem serviços de análise de sentimento como parte de suas ofertas de MLaaS. Exploraremos os serviços desses três provedores nesse trabalho.

2.4 Injeção de Ruído em Dados de Texto

A injeção de ruído em dados de texto é um processo que adiciona aleatoriedade ao texto, seja através da substituição, adição ou exclusão de caracteres, palavras ou mesmo frases. Esta técnica é usada frequentemente para criar cenários de treinamento robustos em aplicações de aprendizado de máquina, incluindo processamento de linguagem natural (NLP) (Wei and Zou, 2019).

No contexto do NLP, a injeção de ruído é particularmente útil para melhorar a robustez dos modelos às variações no *input* (entrada) do usuário (Belinkov and Bisk, 2018). Considerando que os seres humanos são suscetíveis a erros de digitação e que a linguagem natural apresenta variações em termos de gírias, dialetos e normas gramaticais, a injeção de ruído pode ajudar a criar um modelo que seja mais resistente a essas variações.

A injeção de ruído é também uma estratégia eficaz para combater o sobreajuste (Bishop, 2006), um fenômeno comum no aprendizado de máquina em que um modelo treinado aprende o conjunto de treinamento tão bem que tem um desempenho deficiente em dados novos. Ao introduzir ruído no conjunto de treinamento, podemos incentivar o modelo a aprender padrões mais generalizáveis que são menos suscetíveis ao sobreajuste.

Outro benefício importante da injeção de ruído é que ela pode servir como uma forma de aumento de dados (Shorten and Khoshgoftaar, 2019), permitindo que os praticantes ampliem seu conjunto de treinamento sem a necessidade de coletar mais dados. Isto pode ser particularmente importante em situações onde a coleta de dados é cara ou invasiva.

2.4.1 Aplicações

A injeção de ruído tem uma ampla variedade de aplicações no campo do aprendizado de máquina e do NLP. Um exemplo comum de aplicação é na tradução automática. No trabalho de

Belinkov e Bisk (Belinkov and Bisk, 2018), foi demonstrado que tanto o ruído sintético (gerado artificialmente) quanto o natural (erros reais feitos por humanos) podem afetar significativamente a qualidade da tradução automática. Em seu experimento, eles injetaram ruído nos dados de treinamento para tornar o modelo de tradução automática mais robusto a erros de entrada.

A injeção de ruído também é usada na tarefa de classificação de texto. Por exemplo, Xie et al. (Radosavovic et al., 2017) mostraram que a injeção de ruído pode ser usada para criar "dados destilados" que são especialmente eficazes para treinar modelos de aprendizado de máquina.

Outro exemplo de aplicação da injeção de ruído é no campo do reconhecimento de imagem. Na pesquisa médica, por exemplo, a injeção de ruído é usada para simular diferentes condições de imagem e tornar os algoritmos de diagnóstico mais robustos. No trabalho de Irvin et al. (Irvin et al., 2019), foi injetado ruído nas imagens de raios-X para melhorar a capacidade do modelo de lidar com variações nas imagens.

2.5 Tipos de Ruído

A eficácia da injeção de ruído depende em grande parte do tipo de ruído que está sendo injetado. Na literatura sobre aprendizado de máquina e processamento de linguagem natural, vários tipos de ruído têm sido estudados, cada um com suas próprias propriedades e usos. Esta seção fornece uma descrição detalhada de alguns dos tipos de ruído mais comumente usados, e que serão o foco deste estudo. É importante notar que o impacto específico de um determinado tipo de ruído pode variar dependendo do modelo específico e da tarefa de aprendizado de máquina.

2.5.1 Teclado

O ruído do tipo "Teclado" é uma técnica de aumento de dados que simula erros de digitação humanos. Estes erros são frequentemente causados ao pressionar uma tecla adjacente no teclado por acidente. Esta técnica é particularmente útil para melhorar a robustez de modelos de aprendizado de máquina que processam dados de texto que podem conter erros de digitação.

O algoritmo responsável por introduzir este tipo de ruído funciona mapeando cada tecla para suas teclas adjacentes no layout do teclado. Depois, ele substitui caracteres individuais por um caractere que esteja fisicamente próximo no teclado. Por exemplo, se estivéssemos utilizando o layout de um teclado QWERTY padrão e o algoritmo estivesse tentando adicionar ruído ao caracter 's', ele poderia substituí-lo pelos caracteres 'a', 'w', 'd', 'x' ou 'z', pois esses caracteres são adjacente à tecla 's' neste layout de teclado.

Vamos considerar a palavra "casa" como exemplo. Depois de introduzir o ruído do tipo "Teclado", a palavra pode se tornar "caxa", representando um erro de digitação comum onde a tecla 's' é substituída pela tecla adjacente 'x'. Dessa forma, um modelo de aprendizado de máquina treinado com esse tipo de ruído estará melhor preparado para lidar com erros de digitação semelhantes encontrados em dados do mundo real (Xie et al., 2020).

2.5.2 OCR

OCR, sigla em inglês para Reconhecimento Óptico de Caracteres, é uma tecnologia usada para converter diferentes tipos de documentos, como PDFs digitalizados, imagens capturadas por uma câmera digital, ou texto superposto em uma imagem, em dados editáveis e pesquisáveis (Nagy, 2000).

No entanto, os algoritmos OCR não são perfeitos e podem cometer erros durante a conversão de texto em uma imagem para texto digitalizado. Estes erros podem incluir a substituição de caracteres que se parecem (por exemplo, 'o' com '0', ou 'l' com '1'), exclusão de caracteres que não são bem reconhecidos ou a inserção de caracteres extras onde o algoritmo detecta um caractere onde não existe nenhum.

Introduzir ruído do tipo OCR nos dados pode ajudar a simular esses erros e a preparar um modelo de aprendizado de máquina para ser mais robusto em relação a eles. Por exemplo, a palavra "casa" pode ser transformada em "ca5a", representando um erro de OCR onde o caractere 's' é erroneamente reconhecido como '5'. Ao treinar um modelo com esses erros simulados, esperamos que o mesmo tenha melhor desempenho ao processar dados provenientes de OCR (Xu et al., 2021).

2.5.3 Substituição Aleatória de Caracteres

O ruído do tipo "Substituição Aleatória de Caracteres" consiste em alterar caracteres individuais em um texto de maneira aleatória. Este tipo de ruído é interessante para demonstrar o quanto um modelo é afetado pela troca desses caracteres, observando se o mesmo ainda consegue identificar o termo ou significado original do texto mesmo após a inserção do ruído.

Por exemplo, em uma sentença como "O gato pulou sobre o muro", a letra 's' em "sobre" pode ser substituída por 'p', resultando em "O gato pulou pobre o muro". Da mesma forma, a letra 'm' em "muro" pode ser substituída por 'n', resultando em "O gato pulou sobre o nuro". Esses exemplos representam uma alteração no significado das palavras devido a erros de caractere únicos, e que podem alterar significativamente o contexto de um texto, e que consequentemente podem afetar a predição de sentimento.

A implementação deste tipo de ruído em datasets pode auxiliar no desenvolvimento de modelos mais resilientes a essas inconsistências. Ao treinar o modelo com estas variações, a expectativa é que ele consiga generalizar de forma mais eficiente e mantenha um rendimento elevado mesmo ao se deparar com estes ruídos (Xie et al., 2020).

2.5.4 Troca de Palavras

O ruído do tipo "Troca de Palavras" envolve a permutação das palavras adjacentes dentro de uma sentença. Este tipo de ruído simula a variabilidade na ordem das palavras que pode ocorrer na linguagem natural. Por exemplo, em uma frase como "O gato pulou sobre o muro", as

palavras podem ser reordenadas para formar “Pulou o gato sobre o muro”. Embora a nova frase não mantenha a correta gramática da língua portuguesa, ela preserva as palavras-chave e ainda pode transmitir o mesmo significado geral para um humano.

A variação na ordem das palavras é uma característica importante da linguagem natural e pode variar significativamente entre diferentes idiomas e culturas. Modelos de aprendizado de máquina treinados para lidar com esse tipo de ruído podem ser mais robustos a essas variações, resultando em melhor desempenho ao lidar com dados de entrada reais, onde a ordem das palavras pode variar consideravelmente (Belinkov and Bisk, 2018).

No contexto de análise de sentimento, a troca de palavras pode ajudar a criar um modelo mais robusto, já que a opinião ou emoção geral expressa em uma frase pode não depender estritamente da ordem das palavras. Assim, treinar um modelo com esse tipo de ruído pode resultar em um sistema de análise de sentimento mais preciso e robusto (Belinkov and Bisk, 2018).

2.5.5 Divisão de Palavras

O ruído do tipo “Divisão de Palavras” envolve a quebra de palavras individuais em duas ou mais palavras separadas. Por exemplo, uma palavra como “casaco” poderia ser artificialmente dividida em “ca” e “saco”. Este tipo de ruído pode simular erros que podem ocorrer durante o processo de tokenização de texto. A tokenização é um passo importante no processamento de linguagem natural, onde uma string de entrada de texto é dividida em unidades menores, chamadas tokens. Em português e em muitos outros idiomas, um token geralmente corresponde a uma palavra. No entanto, erros de tokenização podem ocorrer, especialmente quando as palavras não estão claramente separadas por espaços ou quando os dados de entrada contêm erros de digitação, abreviações ou palavras compostas.

A introdução deste tipo de ruído nos dados de treinamento pode fazer com que os modelos de aprendizado de máquina sejam mais robustos ao tratar variações ou erros na segmentação de palavras. Isso pode ser particularmente relevante ao trabalhar com textos informais ou não estruturados, como postagens em mídias sociais ou mensagens de texto, onde a gramática e a ortografia nem sempre seguem a norma culta (Belinkov and Bisk, 2018; Kolak et al., 2003).

Assim, ao treinar um modelo para lidar com esse tipo de ruído, podemos melhorar a robustez do modelo para lidar com erros reais de segmentação que podem ocorrer na entrada de dados. Isso é especialmente relevante para a análise de sentimentos, onde a correta segmentação de palavras é crítica para a compreensão do significado e do sentimento expresso em um texto.

2.5.6 Sinônimo

A introdução de ruído do tipo “Sinônimo” consiste em substituir palavras originais por seus sinônimos. O uso de sinônimos para adicionar ruído ao texto é uma técnica frequentemente

empregada para aumentar a robustez dos modelos de aprendizado de máquina à variação lexical. Além disso, essa abordagem pode ajudar a expandir a cobertura do modelo para um maior número de palavras semelhantes, possibilitando uma melhor generalização para novas palavras ou frases que o modelo pode encontrar durante a fase de teste (Araujo et al., 2020).

Por exemplo, considere a seguinte frase: "O clima está muito agradável hoje". A introdução de ruído de sinônimo poderia resultar em uma frase como "O tempo está bastante prazeroso hoje", que mantém um significado semântico semelhante, mas usa palavras diferentes.

A substituição por sinônimos é frequentemente realizada em conjunto com um léxico ou uma ferramenta como word2vec para encontrar sinônimos adequados. Word2vec é um grupo de modelos de aprendizado de máquina relacionados usados para produzir word embeddings. Esses modelos são redes neurais de duas camadas que são treinadas para reconstruir contextos linguísticos de palavras. Eles podem capturar semânticas e sintaxe de palavras e podem ser usados para encontrar palavras que são semelhantes em significado (Devlin et al., 2018; Mikolov et al., 2013a).

A introdução deste tipo de ruído nos dados de treinamento é uma maneira eficaz de melhorar a robustez do modelo, especialmente em tarefas como a análise de sentimento, onde a compreensão de sinônimos e a capacidade de lidar com a variação lexical são cruciais.

2.5.7 Antônimo

A introdução de ruído do tipo "Antônimo" consiste em substituir palavras originais por seus antônimos. Em contraste com o ruído de sinônimo, o ruído de antônimo pode alterar significativamente o significado de uma frase. Apesar disso, existem situações em que esta técnica pode ser útil.

Por exemplo, em certos contextos, pode-se esperar que um modelo de aprendizado de máquina seja capaz de entender e lidar com a negação. Na linguagem natural, a negação é frequentemente expressa através do uso de antônimos. Portanto, a inclusão de ruído de antônimo nos dados de treinamento pode ajudar a melhorar a capacidade do modelo de lidar com frases negativas.

Considere a seguinte frase: "Eu gosto de dias frios". A introdução de ruído de antônimo pode transformar a frase em "Eu detesto dias quentes". Embora o significado global da frase tenha sido invertido, a estrutura sintática da frase e a relação entre as palavras permanecem as mesmas.

Além disso, a introdução de ruído de antônimo pode ser útil para aumentar a robustez de um modelo a mudanças semânticas. Isso pode ser particularmente útil em tarefas como análise de sentimento, onde a compreensão de antônimos pode ser crucial para determinar o sentimento expresso em uma frase (Coulombe, 2018).

2.5.8 Ortografia

A introdução de ruído do tipo "Ortografia" é realizada através da simulação de erros ortográficos comuns. Erros ortográficos ocorrem naturalmente na escrita humana e podem ser causados por uma variedade de fatores, incluindo erros de digitação, falta de conhecimento de ortografia correta, e erros de predição de autocorretor. Este tipo de ruído é relevante na medida em que pode ajudar a aumentar a robustez de modelos de aprendizado de máquina a erros de ortografia que possam ocorrer nos dados de entrada.

Existem várias maneiras específicas de introduzir ruído de ortografia. Algumas abordagens comuns incluem inversões de letras (por exemplo, "clima" torna-se "cilma"), omissões de letras (por exemplo, "clima" torna-se "clma"), inserções de letras (por exemplo, "clima" torna-se "cliima") e substituições de letras (por exemplo, "clima" torna-se "cliva"). É importante observar que a introdução deste tipo de ruído deve ser realizada de uma maneira que produza erros plausíveis de ortografia, que são os tipos de erros que um humano pode fazer.

Erros de ortografia por falta de conhecimento da norma culta costumam depender do idioma, e portanto, a implementação desse tipo de ruído requer que o idioma do texto que está sendo tratado seja considerado. Como exemplo, no Português, podemos ter a expressão "a gente" escrita incorretamente como "agente".

A introdução deste tipo de ruído em dados de treinamento pode ser especialmente útil para tarefas que envolvem a análise de texto gerado por humanos, como análise de sentimento, reconhecimento de entidade nomeada, tradução automática, entre outros (Al Sharou et al., 2021).

2.5.9 Substituição de palavras baseadas em TF-IDF

A técnica TF-IDF, uma abreviação para Term Frequency-Inverse Document Frequency, é uma estatística numérica que visa refletir a importância de uma palavra em um documento dentro de uma coleção ou corpus (Manning et al., 2008). Ela aumenta proporcionalmente ao número de vezes que uma palavra aparece no documento, mas é compensada pela frequência da palavra no corpus, ajudando a ajustar o fato de que algumas palavras aparecem com mais frequência em geral.

Formalmente, podemos definir o valor TF-IDF conforme a seguir:

$$\text{TF-IDF}(t,d,D) = \text{TF}(t,d) \times \text{IDF}(t,D) \tag{2.1}$$

Onde:

- TF(t, d) representa a frequência do termo 't' no documento 'd'. Esta é a Term Frequency (TF).
- IDF(t, D) é o logaritmo inverso da fração do número total de documentos no corpus 'D'

que contêm o termo ‘t’. Esta é a Inverse Document Frequency (IDF).

$$\text{IDF}(t,D) = \log\left(\frac{D}{df(t)}\right) \tag{2.2}$$

Onde ‘df(t)’ é o número de documentos que contêm o termo ‘t’, também conhecido como Document Frequency do termo ‘t’.

No contexto da injeção de ruído, a aplicação da técnica TF-IDF é interessante por várias razões. Primeiro, permite identificar e preservar as palavras-chave mais informativas em um texto durante o processo de injeção de ruído. Isso pode melhorar a robustez e a generalização do modelo de aprendizado de máquina, uma vez que a informação crítica não é perdida durante o processo de injeção de ruído.

Em segundo lugar, a substituição de palavras baseada em TF-IDF pode gerar variações mais ricas dos dados de treinamento, já que substitui palavras de acordo com sua importância relativa. As palavras com baixo score de TF-IDF (ou seja, aquelas que são menos informativas) têm uma maior probabilidade de serem substituídas, proporcionando maior diversidade nos dados de treinamento. Esta diversidade pode melhorar a capacidade do modelo de generalizar em textos não considerados antes da aplicação dessa técnica (Storek et al., 2023).

3

Metodologia

Neste capítulo, apresentamos a base de dados utilizada para o estudo, assim como explicamos o processo pela qual a amostragem de dados foi feita e alguns detalhes de implementação. Também explicamos a métrica utilizada na análise dos resultados, explicando o por que da escolha desta.

3.1 Base de Dados

Para este estudo, optamos pela utilização da base de dados "Twitter US Airline Sentiment" (CrowdFlower, 2015). Esta base de dados é o resultado de um trabalho de análise de sentimento sobre os problemas das grandes companhias aéreas dos EUA. Os dados do Twitter foram coletados em Fevereiro de 2015, e os colaboradores classificaram os tweets como positivos, negativos ou neutros. Posteriormente, eles categorizaram as razões negativas, como "voo atrasado" ou "serviço rude".

A escolha desta base de dados foi motivada por várias razões. Primeiramente, os dados do Twitter são altamente relevantes para a análise de sentimento, pois refletem as opiniões e sentimentos do público em tempo real. Além disso, eles fornecem um panorama das opiniões do cliente sobre várias companhias aéreas, proporcionando uma gama diversificada de sentimentos e tópicos.

Em segundo lugar, esta base de dados é particularmente valiosa por serem classificações realizadas manualmente por seres humanos. Isso garante que as classificações de sentimentos sejam confiáveis e representem com precisão os sentimentos expressos nos tweets. Além disso, a categorização das razões negativas oferece uma visão mais aprofundada dos fatores que contribuem para sentimentos negativos, o que pode ser útil para identificar áreas específicas de melhoria para as companhias aéreas.

Por fim, a escolha desta base de dados também está alinhada com o nosso objetivo de testar a robustez e a confiabilidade dos serviços de MLaaS em cenários do mundo real. Dado que

as empresas estão cada vez mais utilizando análises de sentimento para monitorar as opiniões dos clientes e melhorar seus serviços, a avaliação dos serviços de MLaaS utilizando dados do Twitter pode fornecer insights valiosos sobre o seu desempenho em aplicações reais.

3.2 Processo de Amostragem

Devido às limitações e restrições impostas pelos provedores de serviços de MLaaS, incluindo quotas de serviço e limites diários, optamos por selecionar amostras de tamanho 100 para cada tipo de ruído e cada serviço de MLaaS.

Essa decisão foi motivada tanto pela necessidade de manter as demandas dentro das restrições do serviço, quanto pela vontade de garantir uma amostra representativa dos dados. Uma amostra de 100 registros para cada tipo de registro mostrou-se suficiente para exibir os resultados buscados nos objetivos desse estudo.

Para cada um dos nove tipos de ruído discutidos na fundamentação teórica, uma amostra de tamanho 100 foi selecionada aleatoriamente do conjunto de dados. Assim, para cada tipo de ruído, um total de 900 amostras foram utilizadas para cada serviço de MLaaS, totalizando 2700 amostras para todos os três serviços. Para estabelecer um controle, submetemos também aos serviços MLaaS amostras sem injeção de ruído, de forma a avaliar a performance base de seus modelos e fazer um comparativo na apresentação dos resultados.

A amostragem foi realizada de forma estratificada, garantindo que a proporção de tweets positivos, neutros e negativos em cada amostra refletisse a distribuição no conjunto de dados original. Isso assegura a representatividade das amostras e permite uma comparação mais equitativa entre os serviços de MLaaS.

3.3 Injeção de Ruído nos Dados

Para testar a robustez e confiabilidade dos serviços de análise de sentimentos MLaaS selecionados, introduzimos diferentes ruídos no dataset de tweets mencionado na seção anterior.

Usamos a biblioteca NlpAug (Ma, 2021) para gerar variações de cada tweet no nosso dataset. A NlpAug suporta vários tipos de ruído, como troca de palavras, substituição de sinônimos, divisão de palavras, substituição por antônimos, introdução de erros de ortografia, entre outros descritos na fundamentação teórica.

Além dos ruídos em si, é possível variar o nível dos mesmos, em porcentagem, de forma que mais ou menos partes do texto sejam afetadas. Nos nossos experimentos, testamos ruídos na faixa de 10 a 90%, incrementando em faixas de 10%, trazendo 9 diferentes níveis de ruído. Com os resultados sem injeção de ruídos, formamos 10 diferentes níveis que foram analisados.

3.4 Detalhes da implementação com NlpAug

A biblioteca NlpAug fornece alguns parâmetros configuráveis para cada tipo de ruído. A seguir descrevemos como configuramos estes parâmetros e o significado destes.

3.4.1 *Augmentation* de caracteres

O módulo de *augmentation* de caracteres da NlpAug trata da implementação dos ruídos que envolvem a manipulação caractere por caractere do texto, incluindo o ruído de teclado, OCR e substituição aleatória de caracteres.

Em nossa implementação, o parâmetro *aug_level* define a porcentagem de caracteres que sofrerão alteração no texto.

3.4.2 *Augmentation* de palavras

No módulo de *augmentation* de palavras, a NlpAug implementa os tipos de ruídos que tratam de palavras completas no texto. Nesse módulo estão incluídos todos os outros tipos de ruído tratados neste trabalho que não foram citados na seção anterior.

Nestes tipos de ruído, o parâmetro *aug_p* passado para a NlpAug representa a porcentagem de palavras que o ruído será aplicado.

A seguir esclarecemos detalhes de alguns tipos de ruído que envolvem parâmetros ou dados adicionais para gerar o texto.

Antônimo e Sinônimo

Nos tipos de ruído por substituição de sinônimo e antônimo é necessário especificar o idioma do texto original, para que haja substituição correta das palavras. Em nosso estudo, usamos o idioma inglês, que é o idioma do dataset estudado.

Ortografia

Para a implementação do ruído de ortografia, a NlpAug requer um dicionário com as possíveis substituições para cada palavra. Decidimos escolher a base de dados proveniente do estudo (Belinkov and Bisk, 2018), uma vez que oferece um dicionário robusto com mais de 10.000 termos e suas respectivas substituições com possíveis erros ortográficos.

TF-IDF

A substituição utilizando a abordagem TF-IDF requer um modelo que contém a frequência das palavras no corpus. Para isso, é necessário treinar esse modelo com a base de dados escolhida para o estudo. Em nosso caso, nós usamos o dataset de tweets para esse fim.

A NlpAug fornece o módulo *nlpaug.model.word_stats*, que implementa o treinamento da base de dados e gera o modelo necessário para a aplicação deste tipo de ruído.

3.5 Ferramentas utilizadas

A implementação do estudo foi feita utilizando a linguagem de programação Python, utilizando-se de algumas bibliotecas bem conhecidas na ciência de dados, como o SciKit Learn[1], Pandas[2] e NumPy[3]. Como mostrado nas seções anteriores, usamos a biblioteca NlpAug (Ma, 2021) para injetar os diversos tipos de ruídos em cada um dos tweets. Também utilizamos as bibliotecas de integração dos serviços de MLaaS para facilitar a implementação da comunicação com os mesmos.

Para a visualização dos resultados, foi usada a biblioteca matplotlib[4].

3.6 Métricas de Avaliação

A definição de métricas de avaliação é fundamental para avaliar a performance dos serviços de MLaaS utilizados neste estudo. As métricas fornecem uma medida quantitativa da precisão, eficácia e confiabilidade dos serviços em análise de sentimento, sob a influência de diferentes tipos de ruídos.

3.6.1 Precisão

A Precisão, também conhecida como Valor Preditivo Positivo, é uma métrica que mede a proporção de identificações positivas feitas corretamente. Em outras palavras, é a proporção de previsões positivas (por exemplo, a classe "positiva" em uma classificação binária) que foram realmente corretas.

Formalmente, a Precisão é definida como:

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

Onde TP ("Verdadeiros Positivos") se refere ao número de exemplos corretamente classificados como positivos pelo modelo, enquanto FP ("Falsos Positivos") se referem ao número de exemplos incorretamente classificados como positivos.

A precisão é uma métrica especialmente importante quando os custos de Falsos Positivos são altos. Em outras palavras, quando é mais prejudicial classificar erroneamente um exemplo

[1] https://scikit-learn.org/
[2] https://pandas.pydata.org/
[3] https://numpy.org/
[4] https://matplotlib.org/

negativo como positivo do que classificar erroneamente um exemplo positivo como negativo (Sokolova and Lapalme, 2009).

3.6.2 Sensibilidade (Recall)

A sensibilidade, ou Recall, é a proporção de instâncias positivas verdadeiras (verdadeiros positivos) que foram identificadas corretamente pelo modelo entre todas as instâncias positivas reais (Sokolova and Lapalme, 2009).

Formalmente, a sensibilidade é definida como:

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

Um alto valor de Recall indica que o modelo é bom em detectar as instâncias positivas e tem um baixo erro de falsos negativos. No entanto, é importante observar que um alto valor de Recall não implica necessariamente em uma boa precisão. Isso ocorre porque o modelo pode estar classificando um grande número de instâncias como positivas, incluindo aquelas que são realmente negativas (resultando em muitos falsos positivos), a fim de capturar todas as instâncias positivas reais (Sokolova and Lapalme, 2009).

3.6.3 F-Measure

A F-Measure, também conhecida como F-Score ou F1 Score, é uma métrica que combina Precisão e Recall em uma única medida de desempenho. A F-Measure é a média harmônica de Precisão e Recall, e, portanto, considera ambos os aspectos (Van Rijsbergen, 2013).

Formalmente, a F-Measure é definida como:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.3}$$

O valor da F-Measure varia entre 0 e 1, sendo que 0 reprensenta o pior resultado possível, e 1 representa precisão e recall perfeitos. A F-Measure é especialmente útil em situações onde se deseja balancear precisão e recall e não há uma preferência clara entre os dois. Por exemplo, em tarefas de classificação binária ou multiclasse, como a análise de sentimento, pode ser igualmente importante identificar corretamente as instâncias positivas (alta precisão) e minimizar a omissão de instâncias positivas (alto recall) (Sasaki, 2007).

Optar pela F-Measure ao invés de precisão ou recall pode proporcionar uma visão mais equilibrada do desempenho do modelo. Enquanto um modelo pode ter uma alta precisão, ele pode estar perdendo muitas instâncias positivas verdadeiras, resultando em um baixo recall. Da mesma forma, um modelo com alto recall pode estar incorretamente classificando muitas instâncias negativas como positivas, levando a uma baixa precisão. A F-Measure ajuda a ilustrar o compromisso entre precisão e recall para um modelo específico (Powers, 2011).

Devido a esse aspecto de balancear ambas métricas, optamos por prosseguir com o uso da métrica F-measure nos experimentos e análises apresentados nesse trabalho.

4

Resultados e Discussões

Neste capítulo apresentamos e discutimos os resultados obtidos ao executar os experimentos propostos nesse trabalho. Primeiramente, mostramos os resultados de predição dos serviços da Amazon, Google e Microsoft sem ruído adicionado, de forma a estabelecer um resultado base, em seguida, mostramos e discutimos os resultados quando acrescentados os tipos de ruído discutidos anteriormente.

4.1 Resultados base de análise de sentimento nos provedores

Nesta seção demostramos como cada um dos provedores de análise de sentimento se comportaram na análise de sentimento sem ruído adicional introduzido à base de dados.

Provedor	**F-Measure**
Amazon	0.70
Google	0.73
Microsoft	0.56

Tabela 4.1: F-Measure para os provedores Amazon, Google e Microsoft

Os resultados apresentados na Tabela 4.1 mostram a F-Measure para os três provedores de MLaaS avaliados neste estudo: Amazon, Google e Microsoft. Estes resultados foram obtidos sem a introdução de ruído adicional aos dados, servindo assim como uma linha de base para avaliação do desempenho desses provedores em condições ideais.

O provedor Google apresentou a maior F-Measure, 0.73, indicando um equilíbrio mais robusto entre precisão e recall em suas previsões de análise de sentimento. O provedor Amazon, com uma F-Measure de 0.70, também apresentou um desempenho sólido. O provedor Microsoft, por outro lado, obteve uma F-Measure de 0.56, o que indica um desempenho inferior em relação aos outros dois provedores.

Vale ressaltar que uma maior F-Measure não necessariamente indica que um provedor é "melhor" que outro de maneira absoluta. A F-Measure é uma métrica que busca equilibrar precisão e recall, e diferentes aplicações podem exigir mais foco em uma dessas métricas do que na outra. Por exemplo, em uma aplicação onde é crítico evitar falsos positivos, um provedor com maior precisão pode ser preferível mesmo com uma F-Measure ligeiramente inferior (Sokolova and Lapalme, 2009; Powers, 2011).

4.2 Resultados ao introduzir ruídos

4.2.1 Teclado

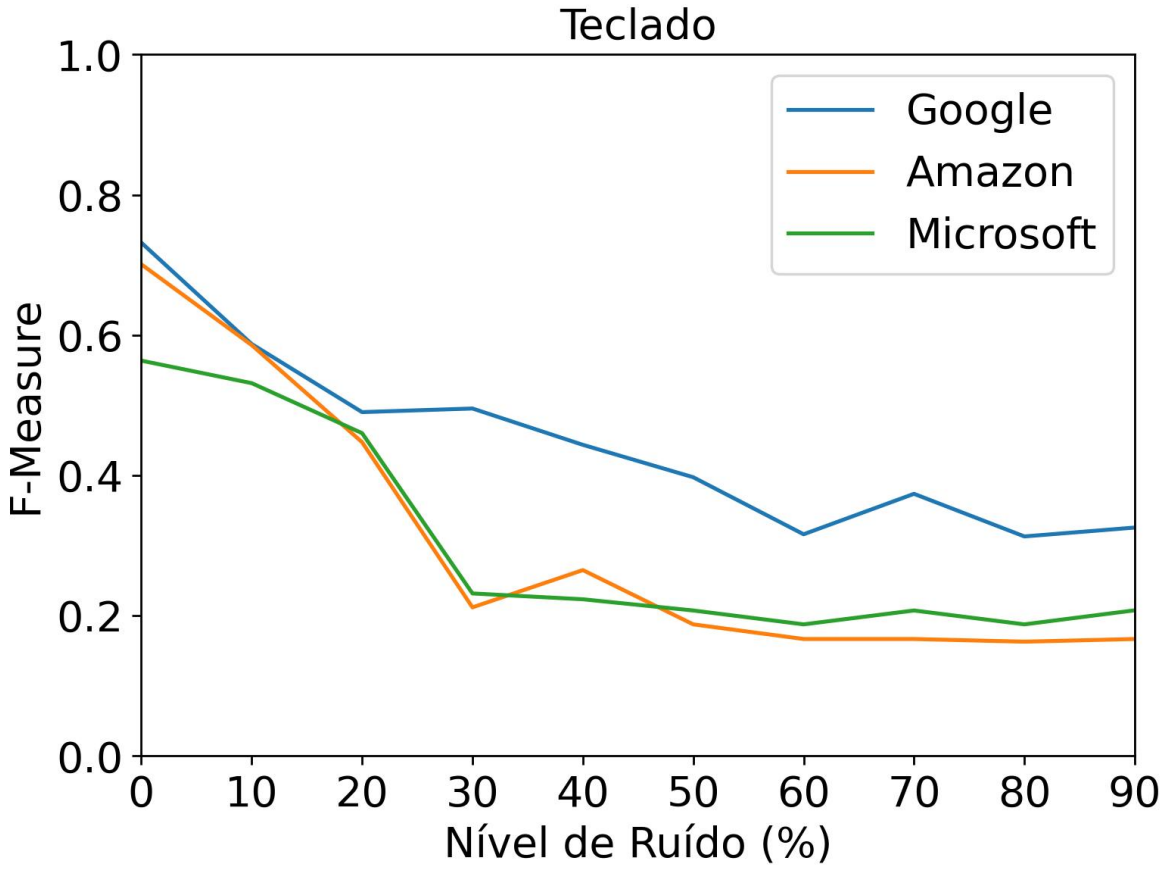

Figura 4.1: Resultados ao aplicar o ruído do tipo teclado

Os resultados apresentados no gráfico da figura 4.1 mostram a F-Measure para os três provedores de MLaaS avaliados (Amazon, Google e Microsoft) sob a introdução de ruído de teclado nos dados, variando de 10% a 90% em incrementos de 10%.

Com o aumento do nível de ruído, todos os três provedores mostraram uma queda no valor da F-Measure. Isso indica uma diminuição na eficácia dos modelos de MLaaS em equilibrar precisão e recall à medida que os dados se tornam mais ruidosos.

Observa-se que, independentemente do nível de ruído, o Google mantém uma F-Measure relativamente alta em comparação com a Amazon e a Microsoft. Isso sugere que o serviço de MLaaS do Google pode ser mais robusto a esse tipo de ruído.

A Amazon, apesar de apresentar uma F-Measure inicial semelhante à do Google, apresenta uma queda mais acentuada à medida que o nível de ruído aumenta, ultrapassando até mesmo a Microsoft em altos níveis de ruído.

A Microsoft, por sua vez, apesar de ter a menor F-Measure inicial, apresenta uma queda menos acentuada com o aumento do ruído, chegando a ultrapassar a Amazon em níveis de ruído muito altos. Isso pode indicar que, embora o serviço de MLaaS da Microsoft possa não ser tão preciso quanto os outros em condições ideais, ele pode ser mais resistente a esse tipo de ruído.

4.2.2 OCR

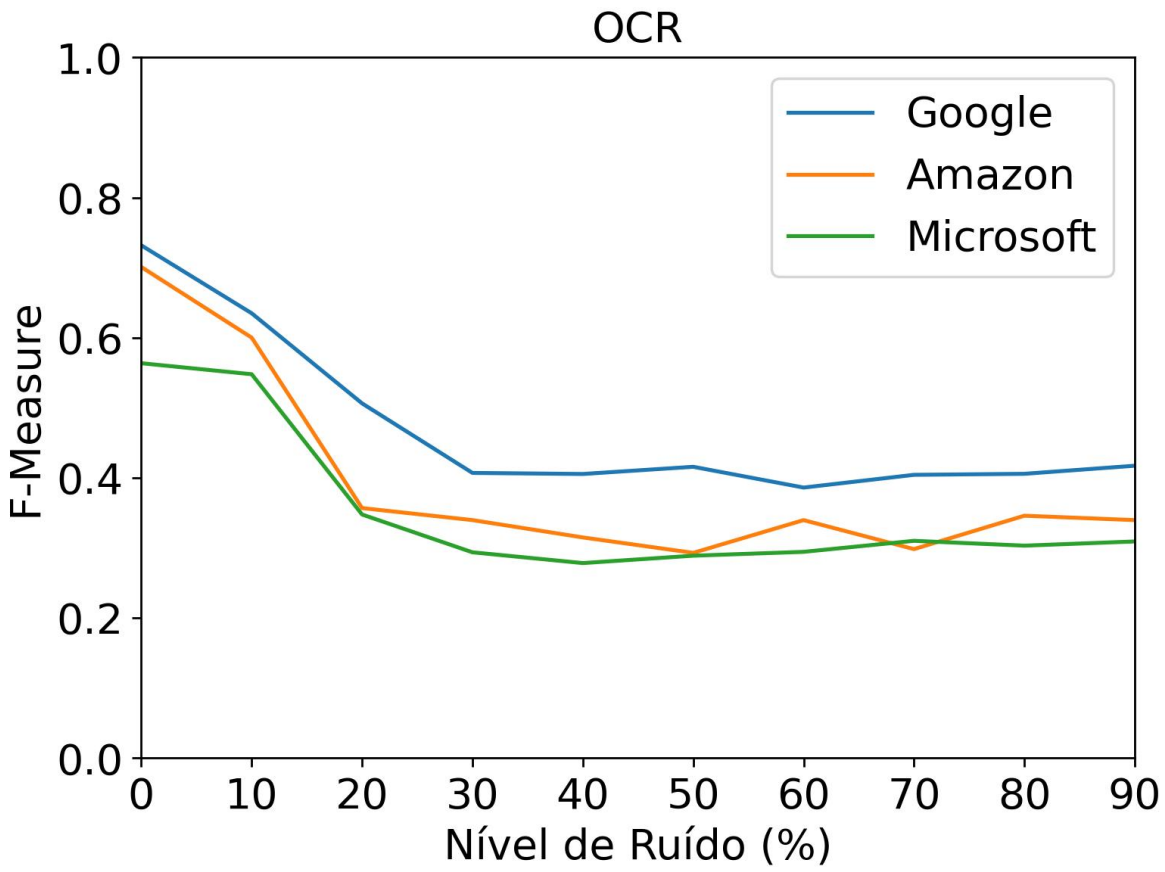

Figura 4.2: Resultados ao aplicar o ruído do tipo OCR

No gráfico da figura 4.2, os resultados para o tipo de ruído OCR são apresentados. O provedor Google exibe um desempenho superior em relação aos outros, mantendo uma F-Measure consistente mesmo em altos níveis de ruído. Isso poderia ser atribuído a estratégias de treinamento

robustas e a modelos mais complexos que são capazes de capturar a variabilidade nos dados mesmo em condições adversas (Goodfellow et al., 2016).

A Amazon, por outro lado, apresenta uma queda significativa na F-Measure conforme o nível de ruído aumenta. Isso poderia ser indicativo de que o modelo de análise de sentimento da Amazon pode ser mais sensível a variações e ruídos nos dados. Tais variações podem ser mais difíceis de capturar e podem levar a um desempenho inferior em cenários do mundo real onde os dados são naturalmente ruidosos e variáveis (Provost and Fawcett, 2013).

Em relação à Microsoft, apesar de seu desempenho inicial ser mais baixo, sua queda na F-Measure é mais gradual e menos severa do que a da Amazon. Isso pode ser um indicativo de que o modelo de análise de sentimento da Microsoft pode ter uma maior resistência a ruídos. Tal resistência pode ser uma consequência de uma estratégia de treinamento mais regularizada que evita overfitting e, portanto, pode ser mais resistente a variações nos dados (Krogh and Hertz, 1992; Srivastava et al., 2014).

4.2.3 Substituição Aleatória de Caracteres

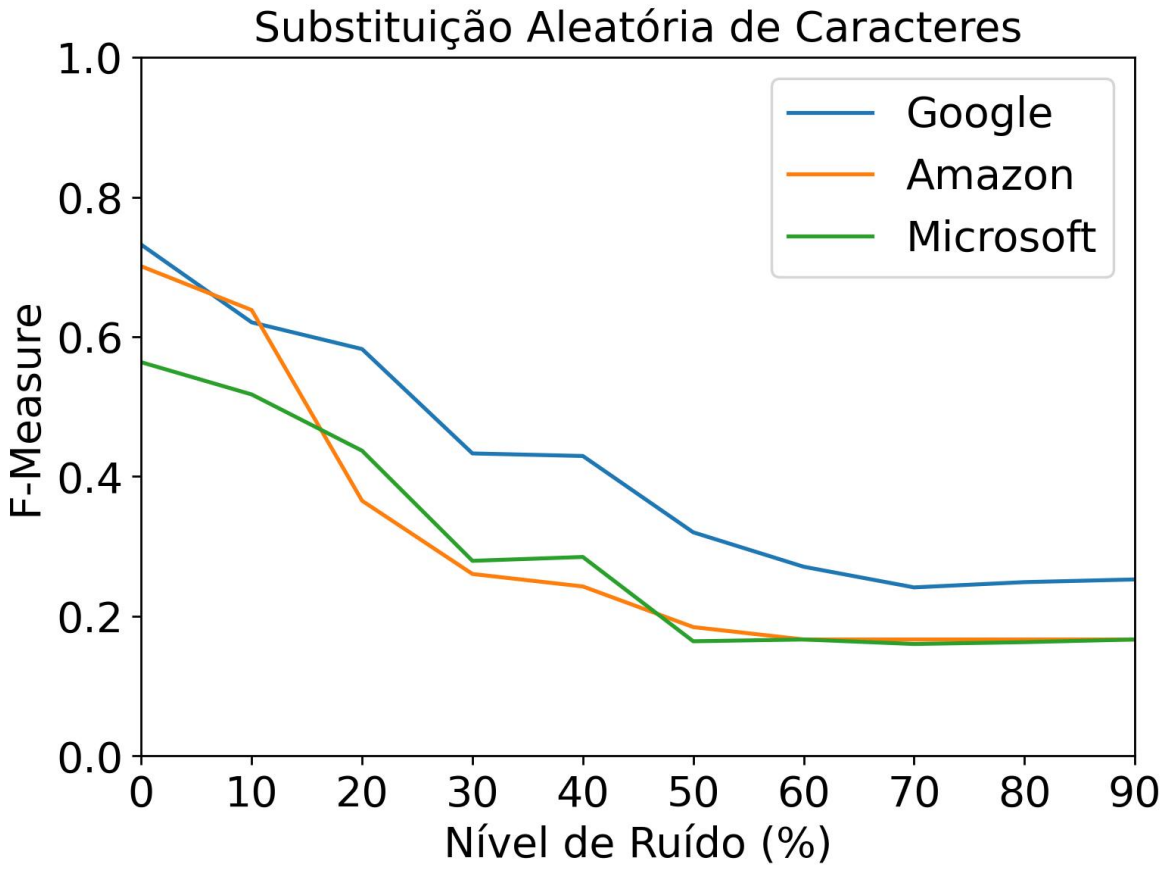

Figura 4.3: Resultados ao aplicar o ruído de substituição aleatória de caracteres

No gráfico da figura 4.3 temos o gráfico representando os resultados do ruído de substituição aleatória de caracteres. Os resultados são em grande parte semelhantes aos obtidos para os ruídos de teclado e OCR, tendo o Google mantido um desempenho superior, enquanto a solução da Amazon sofreu uma queda brusca nos níveis iniciais de ruído, estabilizando em seguida.

Nota-se que a F-Measure estabiliza-se em altos níveis de ruído, assim como o fez nos ruídos anteriores, talvez por o significado do texto já ter sido completamente perdido.

4.2.4 Troca de Palavras

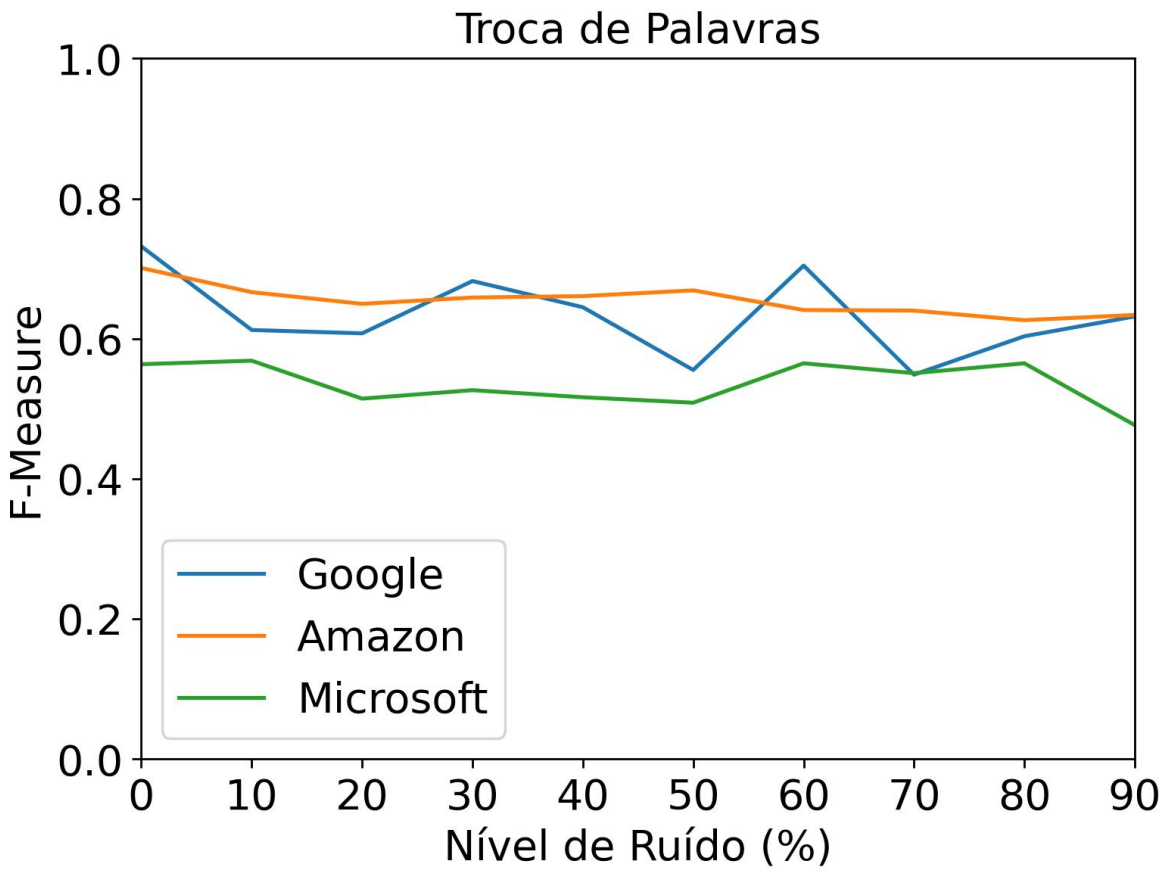

Figura 4.4: Resultados ao aplicar o ruído de Troca de Palavras

No gráfico da figura 4.4, são apresentados os resultados obtidos para o tipo de ruído de troca de palavras. Notamos que os valores da F-Measure permanecem razoavelmente estáveis para todos os três provedores, mesmo com o aumento do nível de ruído.

Esse resultado pode ser atribuído à natureza do tipo de ruído em questão. A troca de palavras, ao contrário da substituição de caracteres ou inserção de ruído aleatório, preserva uma quantidade significativa de informação no texto. Isso ocorre porque, mesmo quando palavras são trocadas, o contexto geral do texto pode ser parcialmente mantido devido à presença de outras palavras e estruturas gramaticais que permanecem intactas.

Isso poderia explicar por que os modelos foram capazes de manter um desempenho relativamente estável, mesmo em altos níveis de ruído. Tais modelos, especialmente os que utilizam técnicas avançadas de NLP como mecanismos de atenção e transformação, são eficazes na captura de informações contextuais em textos, o que poderia ajudá-los a resistir a esse tipo de ruído (Vaswani et al., 2017; Devlin et al., 2018).

Adicionalmente, é relevante notar que a Amazon e o Google mostram uma variação maior na F-Measure em comparação à Microsoft. Esta última, apesar de começar com um valor de F-Measure mais baixo, exibe uma variação menor em resposta ao aumento do ruído. Isso poderia indicar que o modelo da Microsoft é possivelmente mais robusto a esse tipo de ruído, ou que é mais capaz de se adaptar a mudanças na composição dos dados.

4.2.5 Divisão de Palavras

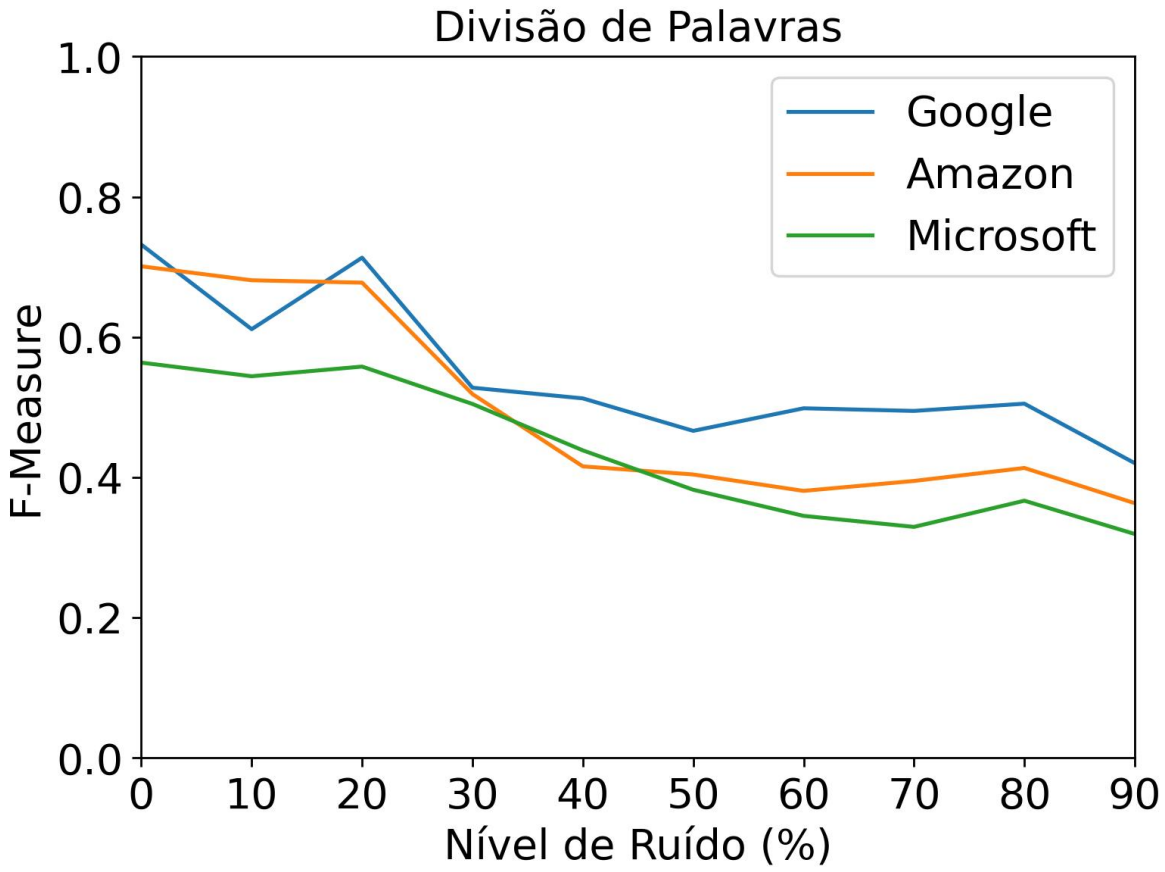

Figura 4.5: Resultados ao aplicar o ruído de Divisão de Palavras

O gráfico da figura 4.5 mostra os resultados da aplicação do ruído de troca de palavras. Pode-se observar uma queda gradual na F-Measure para todos os três provedores à medida que o nível de ruído aumenta.

A troca de palavras introduz uma perturbação no nível da informação semântica. Isso significa que o sentido geral do texto pode ser significativamente alterado, tornando a tarefa de análise de sentimentos consideravelmente mais desafiadora. Isso poderia explicar a queda observada na F-Measure à medida que o nível de ruído aumenta (Goodfellow et al., 2016).

A Amazon e o Google começam com uma F-Measure mais alta, mas mostram uma queda mais acentuada à medida que o nível de ruído aumenta. Isso pode indicar que seus modelos podem ser mais sensíveis a perturbações no nível da semântica do texto.

Por outro lado, a Microsoft, apesar de começar com uma F-Measure mais baixa, mostra uma queda mais gradual à medida que o nível de ruído aumenta. Isso pode sugerir que seu modelo de MLaaS é relativamente mais robusto ao ruído de troca de palavras, possivelmente devido à forma como o modelo é treinado para lidar com a variabilidade nos dados, assim como foi observado nos resultados anteriores.

4.2.6 Sinônimo

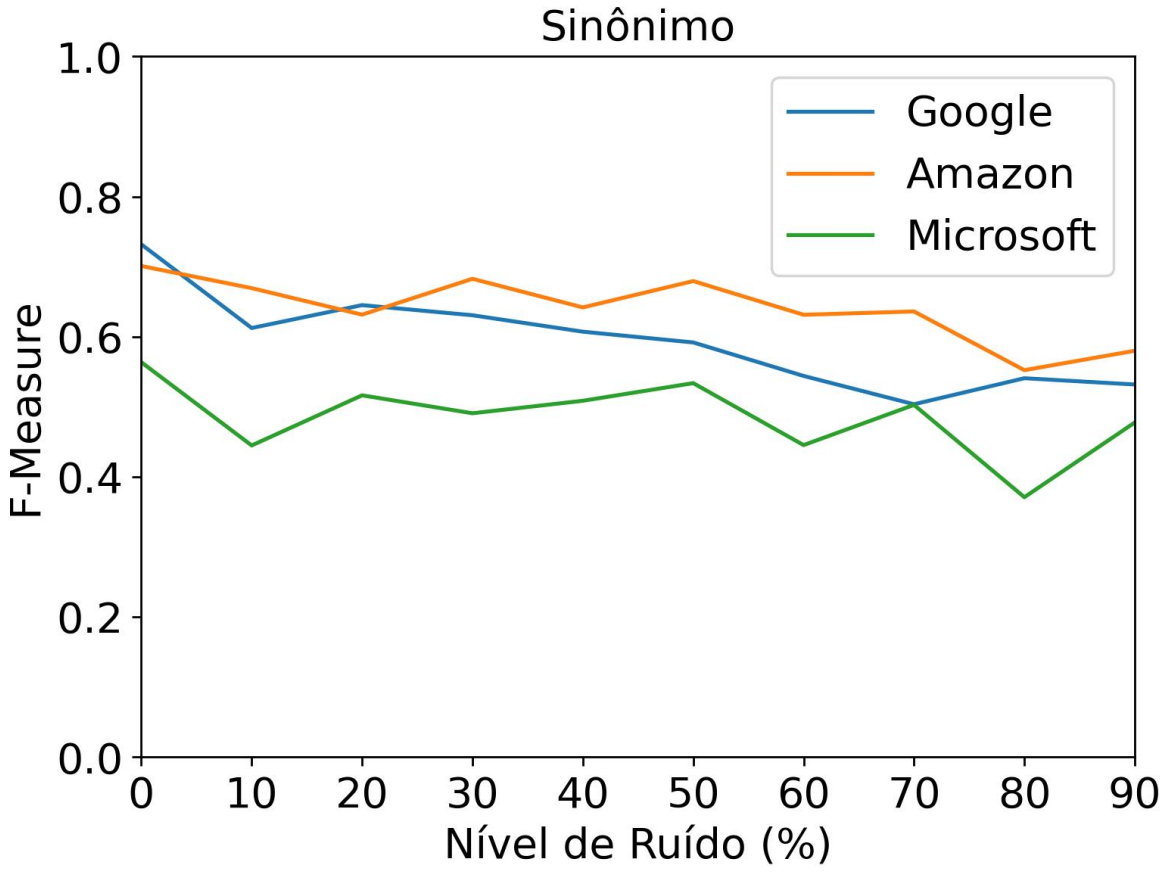

Figura 4.6: Resultados ao aplicar o ruído do tipo Sinônimo

Os resultados para a introdução do ruído de troca de palavras por sinônimos estão ilustrados no gráfico da figura 4.6. Neste cenário, observa-se um comportamento interessante.

A Amazon e a Microsoft mostram uma F-Measure variando ao longo dos diferentes níveis de ruído, sem uma tendência de queda ou subida consistente. Isso pode ser devido ao fato de que, ao trocar palavras por seus sinônimos, a semântica geral do texto é mantida. Portanto, a performance do modelo tende a ser menos afetada por esse tipo de ruído, comparado a outros tipos de perturbações mais disruptivas, como a substituição aleatória de caracteres (Mikolov et al., 2013b).

Por outro lado, o Google mostra uma leve tendência de queda com o aumento do nível de ruído. Embora a troca por sinônimos preserve o sentido geral do texto, existem nuances semânticas entre diferentes sinônimos que podem não ser totalmente capturadas pelo modelo. Isso poderia explicar a ligeira diminuição na F-Measure para o Google (Mikolov et al., 2013a; Řehůřek and Sojka, 2010).

Em resumo, a troca de palavras por sinônimos introduz um tipo de ruído que altera o texto, mas mantém grande parte da informação semântica intacta. Isso faz com que os modelos dos provedores tenham um desempenho relativamente estável neste tipo de cenário, evidenciando sua capacidade de capturar a semântica do texto de maneira eficaz.

4.2.7 Antônimo

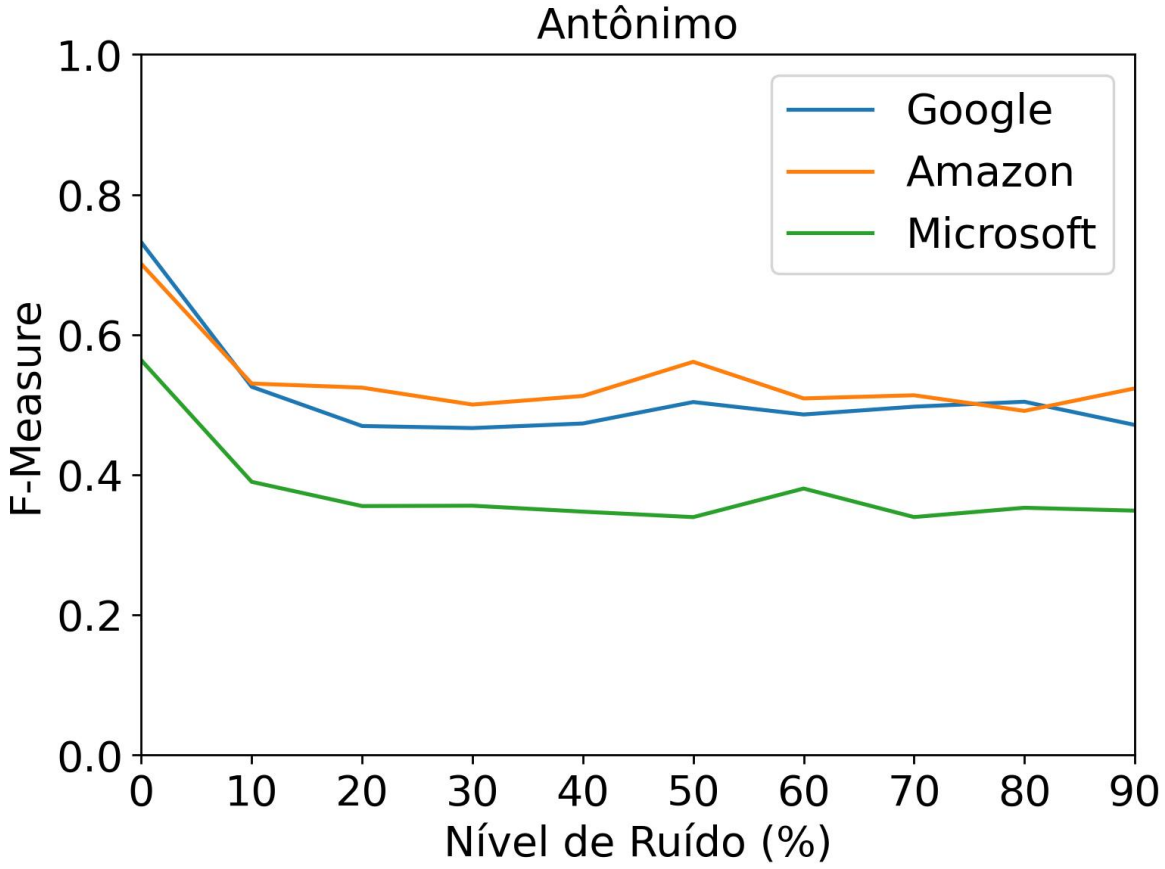

Figura 4.7: Resultados ao aplicar o ruído do tipo Antônimo

No gráfico da figura 4.7, apresentamos os resultados da aplicação de ruído de antônimo, uma tarefa de alta complexidade, pois a substituição de palavras por seus antônimos altera significativamente o significado das sentenças, embora mantenha a estrutura gramatical inalterada.

Os três provedores de MLaaS, Amazon, Google e Microsoft, apresentam uma queda substancial na F-Measure já no primeiro nível de ruído. Esta queda abrupta pode ser justificada pela complexidade em manter a consistência semântica do texto após a substituição por antônimos. Essa mudança de contexto pode confundir os modelos, levando a uma diminuição no desempenho, conforme estudos sugerem (Feng et al., 2019).

O Google, apesar da queda inicial, estabiliza a F-Measure com o aumento do nível de ruído, sugerindo um certo nível de robustez de seu modelo a essas mudanças de contexto.

A Amazon e a Microsoft apresentam o mesmo comportamento do Google, embora com estabilidade ligeiramente menor ao observado no Google. Ainda sim, indicam uma certa resistência à inversão semântica introduzida pelo ruído de antônimos(Michel et al., 2018).

Mesmo com a queda acentuada inicial, os modelos conseguem estabilizar sua performance, sugerindo que, após um certo ponto, as alterações adicionais têm um impacto reduzido.

4.2.8 Ortografia

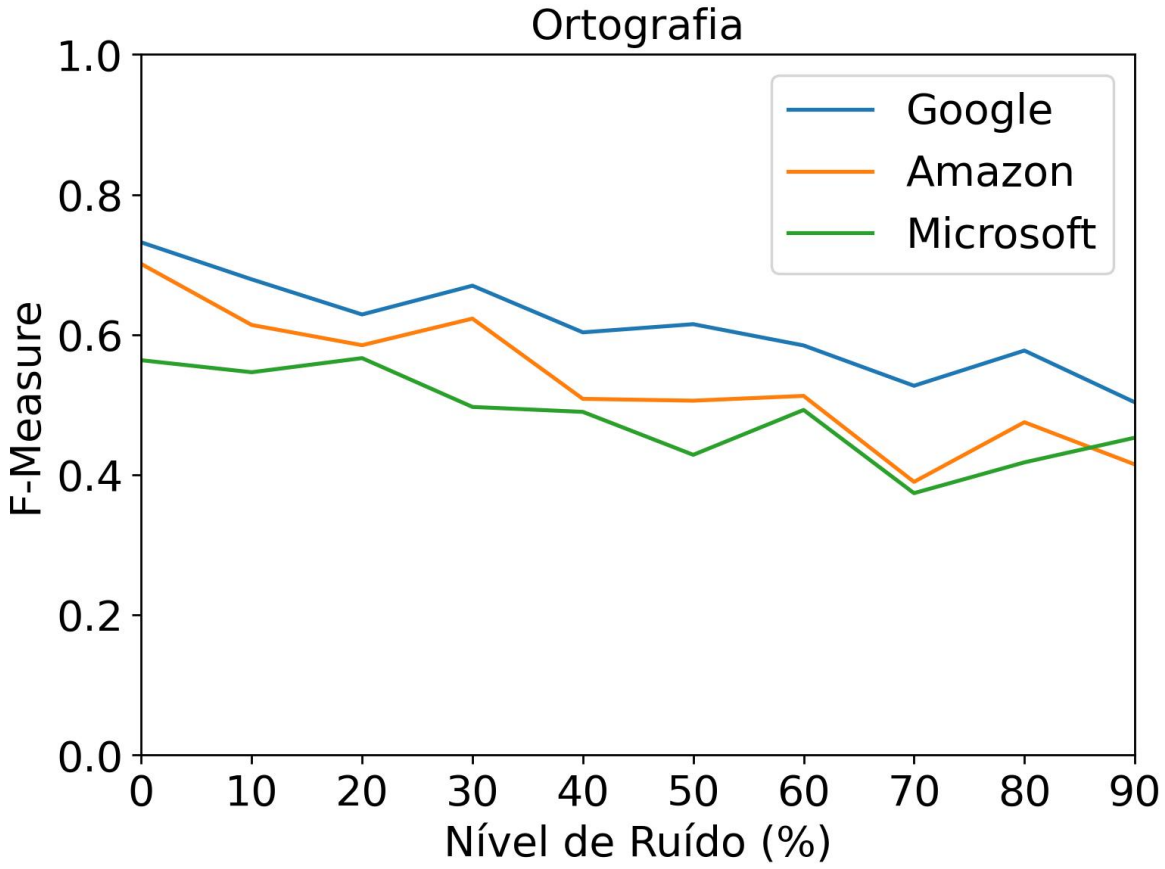

Figura 4.8: Resultados ao aplicar o ruído de Ortografia

Ao contrário dos ruídos de substituição aleatória de caracteres, teclado e OCR, os ruídos de ortografia apresentam uma queda de F-Measure consideravelmente mais branda em todos os três provedores, como mostrado no gráfico da figura 4.8.

Esse fenômeno pode estar associado à natureza deste tipo de ruído. Em ruídos de ortografia, as alterações tendem a ser semanticamente similares às palavras originais e, em muitos casos, as palavras erradas podem ser palavras reais. Estes resultados mostram também um possível enfoque nos modelos a erros de ortografia cometido durante a escrita de texto por humanos.

Com relação aos resultados individuais, o Google mantém uma F-Measure superior, sugerindo que seus modelos podem ter estratégias mais eficazes para lidar com erros de ortografia. A Amazon e a Microsoft mostram uma tendência semelhante de queda de F-Measure, embora a Amazon tenha uma queda mais acentuada nos níveis de ruído mais altos, indicando que seus modelos podem ser mais sensíveis a esses erros.

4.2.9 Substituição de palavras por TF-IDF

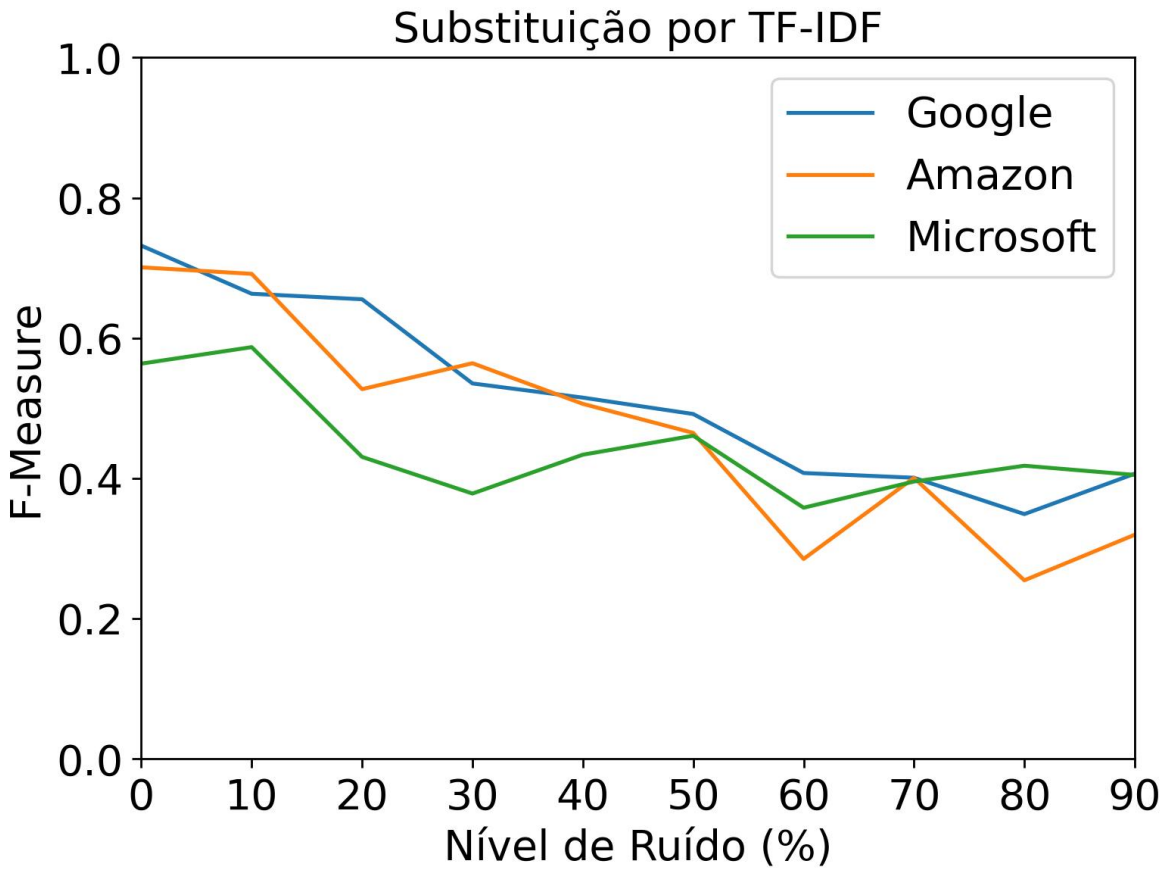

Figura 4.9: Resultados ao aplicar o ruído de substuição de palavras por TF-IDF

Os resultados para o tipo de ruído de substituição por TF-IDF, mostrados no gráfico da figura 4.9, sugerem que os modelos são sensíveis mesmo à modificação de palavras classificadas

como menos relevantes. Em baixo nível de ruído (10%) não há queda ou existe pouca queda na F-Measure, provavelmente devido ao número baixo de palavras (ainda que de baixa importância) substituídas. Entretanto, ao aumentar o nível de ruído e substituir cada vez mais palavras, a F-Measure também cai, o que pode indicar que não apenas as palavras mais relevantes conforme a classificação TF-IDF são relevantes para os modelos MLaaS, mas também a estrutura e restante do texto.

É interessante notar que o modelo da Amazon parece ser mais afetado nos níveis de ruído mais altos, mostrando uma queda significativa na F-measure em alguns dos pontos. Uma explicação possível é que o modelo da Amazon pode ser mais sensível às palavras com baixa importância TF-IDF. Em altos níveis de ruído, quando muitas dessas palavras são substituídas, o modelo pode ter mais dificuldade em entender o significado do texto, resultando em uma menor F-measure.

4.3 Análise Geral

4.3.1 Medianas da F-Measure para cada provedor

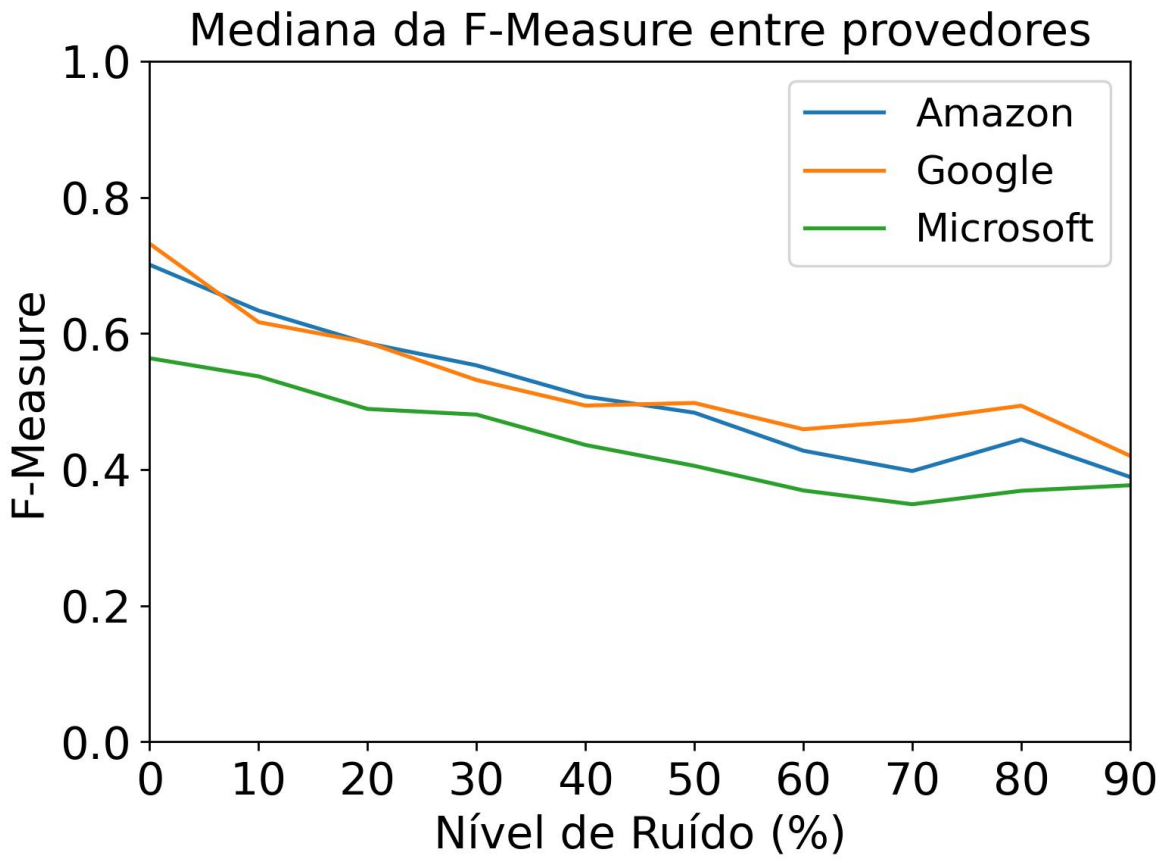

Figura 4.10: Medianas da F-Measure para cada provedor por nível de ruído

O gráfico da figura 4.10 mostra a mediana das F-measures para cada provedor em diferentes níveis de ruído. É notável uma tendência de queda linear para todos os três provedores à medida que o nível de ruído aumenta. Esta tendência sugere que a presença de ruído nos dados de entrada afeta negativamente o desempenho dos modelos de análise de sentimento, como queríamos demonstrar nesse trabalho.

A queda linear sugere que cada incremento no nível de ruído causa uma diminuição proporcional na F-measure. Isso indica que, independente do tipo de ruído introduzido, existe uma relação consistente entre o aumento do ruído e a degradação da performance dos modelos. Isto é, os modelos parecem ser sensíveis tanto a pequenos quanto a altos níveis de ruído, e a capacidade de lidar com ruído não parece melhorar ou piorar de forma muito significativa em níveis de ruído específicos. Embora hajam variações, a tendência geral é de queda, e experimentos com mais disponibilidade de acesso aos provedores aqui estudados poderiam fazer experimentos com datasets e amostras maiores para verificar se essas pequenas variações permanecem.

A Amazon e Google mostram um desempenho ligeiramente melhor em relação à Microsoft. No entanto, todos os provedores demonstram uma redução significativa na F-measure com o aumento dos níveis de ruído. Isso ressalta a importância de estratégias robustas de pré-processamento e limpeza de dados na implementação de soluções de machine learning (Hastie et al., 2009).

4.3.2 Resumo dos Resultados

Tabela 4.2: Variação da F-Measure em relação ao Nível de Ruído (%)

		Nível de Ruído (%)									
		0	10	20	30	40	50	60	70	80	90
Teclado	Amazon	0.70	0.59	0.45	0.21	0.26	0.19	0.17	0.17	0.16	0.17
	Google	0.73	0.59	0.49	0.50	0.44	0.40	0.32	0.37	0.31	0.33
	Microsoft	0.56	0.53	0.46	0.23	0.22	0.21	0.19	0.21	0.19	0.21
OCR	Amazon	0.70	0.60	0.36	0.34	0.31	0.29	0.34	0.30	0.35	0.34
	Google	0.73	0.63	0.51	0.41	0.41	0.42	0.39	0.40	0.41	0.42
	Microsoft	0.56	0.55	0.35	0.29	0.28	0.29	0.29	0.31	0.30	0.31
Substituição Aleatória de Caracteres	Amazon	0.70	0.64	0.37	0.26	0.24	0.18	0.17	0.17	0.17	0.17
	Google	0.73	0.62	0.58	0.43	0.43	0.32	0.27	0.24	0.25	0.25
	Microsoft	0.56	0.52	0.44	0.28	0.28	0.16	0.17	0.16	0.16	0.17
Troca de Palavras	Amazon	0.70	0.67	0.65	0.66	0.66	0.67	0.64	0.64	0.63	0.63
	Google	0.73	0.61	0.61	0.68	0.64	0.56	0.70	0.55	0.60	0.63
	Microsoft	0.56	0.57	0.51	0.53	0.52	0.51	0.56	0.55	0.56	0.48
Divisão de Palavras	Amazon	0.70	0.68	0.68	0.52	0.42	0.40	0.38	0.39	0.41	0.36
	Google	0.73	0.61	0.71	0.53	0.51	0.47	0.50	0.49	0.51	0.42
	Microsoft	0.56	0.54	0.56	0.50	0.44	0.38	0.35	0.33	0.37	0.32
Sinônimo	Amazon	0.70	0.67	0.63	0.68	0.64	0.68	0.63	0.64	0.55	0.58
	Google	0.73	0.61	0.65	0.63	0.61	0.59	0.54	0.50	0.54	0.53
	Microsoft	0.56	0.44	0.52	0.49	0.51	0.53	0.45	0.50	0.37	0.48
Antônimo	Amazon	0.70	0.53	0.52	0.50	0.51	0.56	0.51	0.51	0.49	0.52
	Google	0.73	0.53	0.47	0.47	0.47	0.50	0.49	0.50	0.50	0.47
	Microsoft	0.56	0.39	0.36	0.36	0.35	0.34	0.38	0.34	0.35	0.35
Ortografia	Amazon	0.70	0.61	0.58	0.62	0.51	0.51	0.51	0.39	0.48	0.41
	Google	0.73	0.68	0.63	0.67	0.60	0.61	0.58	0.53	0.58	0.50
	Microsoft	0.56	0.55	0.57	0.50	0.49	0.43	0.49	0.37	0.42	0.45
Substituição por TF-IDF	Amazon	0.70	0.69	0.53	0.56	0.51	0.46	0.29	0.40	0.25	0.32
	Google	0.73	0.66	0.66	0.54	0.52	0.49	0.41	0.40	0.35	0.41
	Microsoft	0.56	0.59	0.43	0.38	0.43	0.46	0.36	0.40	0.42	0.40

Na tabela 4.2 mostramos um resumo dos resultados obtidos nos experimentos executados no decorrer desse trabalho, na tabela, cores de fundo mais próximas do vermelho indicam pior F-Measure. É interessante notar as F-Measures nos níveis de ruido mais altos com os três

primeiros tipos de ruído (Teclado, OCR e Substituição Aleatória de Caracteres), consideravelmente piores em comparação aos outros tipos de ruído nos três provedores.

Essa discrepância nos valores de F-Measure pode ser atribuída ao fato de que esses ruídos podem alterar significativamente a estrutura do texto original, introduzindo erros que podem dificultar a compreensão do conteúdo textual para os modelos. Em particular, a substituição aleatória de caracteres pode levar à formação de palavras que não existem, enquanto erros de teclado e OCR podem criar palavras que são foneticamente semelhantes às originais, mas que têm significados completamente diferentes. Dessa forma, é plausível que os modelos dos provedores tenham dificuldade para processar efetivamente essas formas de ruído textual (Peters et al., 2018; Allen and Hospedales, 2019).

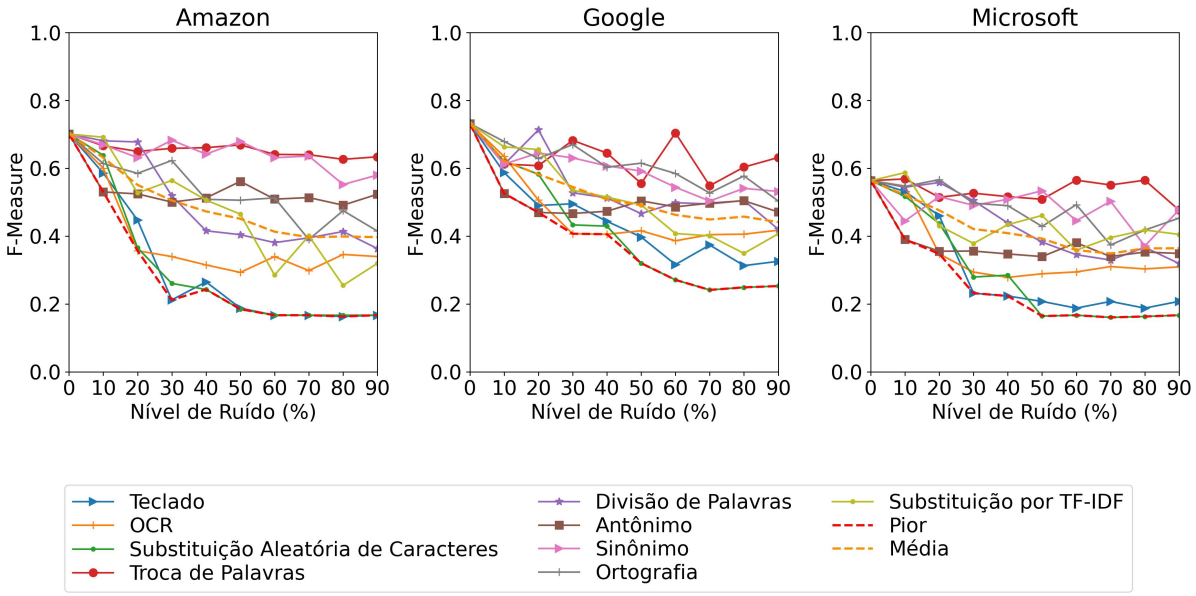

Figura 4.11: Comparação dos tipos de ruído

No gráfico da figura 4.11 mostramos uma visualização condensada de todos os tipos de ruído para cada provedor, também é mostrado uma linha para o pior desempenho em cada nível de ruído, e outra mostrando a média.

5

Conclusão

Neste trabalho, exploramos a performance dos serviços de Machine Learning as a Service (MLaaS) da Microsoft, Amazon e Google no que diz respeito à análise de sentimentos com a presença de ruídos nos dados. Nosso estudo revelou insights valiosos sobre a robustez e confiabilidade desses serviços no tratamento de dados de texto com ruído.

Um ponto fundamental observado durante nossa análise foi o impacto diferencial causado por distintos tipos de ruídos. Descobrimos que ruídos como Teclado, OCR e Substituição Aleatória de Caracteres resultam em um declínio mais acentuado na performance dos modelos. Isso ocorre porque esses tipos de ruídos tendem a alterar drasticamente a estrutura original do texto, podendo criar palavras inexistentes ou modificar o significado original de palavras existentes.

Nossos resultados reforçam a importância do pré-processamento e limpeza dos dados ao implementar soluções de machine learning, evidenciando que a robustez dos serviços de MLaaS na manipulação de dados com ruídos é uma consideração essencial.

Este estudo também abre portas para a realização de trabalhos futuros. Para expandir o conhecimento sobre a robustez dos serviços de MLaaS, recomenda-se a realização de estudos adicionais considerando uma gama maior de provedores, a exploração de outros tipos de ruído e expansão na análise dos ruídos apresentados nesse trabalho, utilizando mais recursos proporciados pela NlpAug. Além disso, a pesquisa pode ser estendida para outros tipos de dados, como imagens ou áudios, bem como para outras tarefas de aprendizado de máquina, além da análise de sentimentos. Também seria interessante investigar diferentes estratégias de pré-processamento de dados e suas eficácias na mitigação dos efeitos do ruído nos serviços de MLaaS.

Em última análise, embora tenhamos obtido descobertas significativas sobre a capacidade dos serviços de MLaaS de lidar com ruído em dados de texto, também identificamos várias oportunidades para pesquisas futuras que permitirão uma compreensão ainda mais profunda desses serviços em diversas aplicações.

Referências bibliográficas

Khetam Al Sharou, Zhenhao Li, and Lucia Specia. Towards a better understanding of noise in natural language processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 53–62, Held Online, September 2021. INCOMA Ltd. URL https://aclanthology.org/2021.ranlp-1.7.

Carl Allen and Timothy Hospedales. Analogies explained: Towards understanding word embeddings. *International Conference on Machine Learning*, 2019.

Amazon. Amazon comprehend, 2023. URL https://aws.amazon.com/comprehend/.

Vladimir Araujo, Andres Carvallo, Carlos Aspillaga, and Denis Parra. On adversarial examples for biomedical nlp tasks, 2020.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*, 2018.

Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Prashant Chaudhary, Pavan Kurariya, Shashi Pal Singh, Jahnavi Bodhankar, Lenali Singh, and Ajai Kumar. Intelligent virtual research environment for natural language processing (ivre-nlp). In Yu-Dong Zhang, Tomonobu Senjyu, Chakchai So-In, and Amit Joshi, editors, *Smart Trends in Computing and Communications*, pages 453–465, Singapore, 2023. Springer Nature Singapore. ISBN 978-981-16-9967-2.

Hongge Chen. *Robust Machine Learning Models and Their Applications*. PhD thesis, 2021.

Claude Coulombe. Text data augmentation made simple by leveraging nlp cloud apis, 2018.

CrowdFlower. Twitter us airline sentiment. https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment, 2015. Accessed: 2023-06-16.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Shaowei Feng, Jun Li, C. Lawrence Zitnick, Anitha Kannan, and Yejin Gan. Keep calm and switch on! preserving sentiment and fluency in semantic text exchange. *Advances in Neural Information Processing Systems*, 32, 2019.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.

Google. Google cloud natural language api, 2023. URL https://cloud.google.com/natural-language.

Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media, 2009.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *arXiv preprint arXiv:1901.07031*, 2019.

Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

Daniel Jurafsky and James H Martin. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall, 2009.

Okan Kolak, Bill Byrne, and Philip Resnik. A generative probabilistic ocr model for nlp applications. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 134–141, 2003.

Anders Krogh and Jesper A Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4:950–957, 1992.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv:1611.01236*, 2016.

Bing Liu. Sentiment analysis and opinion mining. 2012.

Edward Ma. Nlpaug. 2021. URL https://github.com/makcedward/nlpaug.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

Jonas Michel, Ryan L. Hicks, Chu Song, Nicholas Carlini, Wesley Melicher, Vipul Puri, Andrew Norton, Alexey Kurakin, Tadayoshi Kohno, Sascha Fahl, et al. Adversarial examples that fool both computer vision and time-limited humans. *Advances in Neural Information Processing Systems*, 31, 2018.

Microsoft. Azure cognitive services, 2023. URL https://azure.microsoft.com/services/cognitive-services/.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Workshop at the International Conference on Learning Representations*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

Tom M Mitchell. *Machine learning*. McGraw Hill, 1997. ISBN 0070428077. URL http://www.cs.cmu.edu/~tom/mlbook.html.

George Nagy. Twenty years of document image analysis in pami. *IEEE transactions on pattern analysis and machine intelligence*, 22(1):38–62, 2000.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

David M Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

Foster Provost and Tom Fawcett. *Data science for business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media, Inc., 2013.

Ilija Radosavovic, Piotr Dollár, Ross B. Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. *CoRR*, abs/1712.04440, 2017. URL http://arxiv.org/abs/1712.04440.

Mauro Ribeiro, Katarina Grolinger, and Miriam A.M Capretz. Mlaas: Machine learning as a service. *IEEE*, 2015.

Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

Yasumasa Sasaki. The truth of the f-measure. *Teach Tutor mater*, 1(5):1–4, 2007.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 614–622, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. **DOI** 10.1145/1401890.1401965. URL https://doi.org/10.1145/1401890.1401965.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

Adam Storek, Melanie Subbiah, and Kathleen McKeown. Unsupervised selective rationalization with noise injection, 2023.

José A. Sáez, Julián Luengo, and Francisco Herrera. Evaluating the classifier behavior with noisy data considering performance and robustness: The equalized loss of accuracy measure. *Neurocomputing*, 176:26–35, 2016. ISSN 0925-2312. **DOI** https://doi.org/10.1016/j.neucom.2014.11.086. URL https://www.sciencedirect.com/science/article/pii/S0925231215005500. Recent Advancements in Hybrid Artificial Intelligence Systems and its Application to Real-World Problems.

Cornelis Joost Van Rijsbergen. *Information retrieval*. Butterworth-Heinemann, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Jason W Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, 2019.

Cihang Xie, Yuxin Wu, Laurens van der Ma, John C Duchi, and Aleksander Madry. Adversarial examples for non-parametric methods: Attacks, defenses and large sample limits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8510–8518, 2020.

Guowei Xu, Wenbiao Ding, Weiping Fu, Zhongqin Wu, and Zitao Liu. Robust learning for text classification with multi-source noise simulation and hard example mining. In Yuxiao Dong, Nicolas Kourtellis, Barbara Hammer, and Jose A. Lozano, editors, *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track*, pages 285–301, Cham, 2021. Springer International Publishing. ISBN 978-3-030-86517-7.

Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50. ELRA, 2010.