



UNIVERSIDADE
FEDERAL DE
ALAGOAS



FEDERAL UNIVERSITY OF ALAGOAS
INSTITUTE OF COMPUTING
GRADUATE PROGRAM IN INFORMATICS

Masters dissertation

**Evaluating the Resilience of Cloud NLP Services across
Amazon, Microsoft, and Google**

Juliano Rocha Barbosa

jrb@ic.ufal.br

Advisor:

Baldoino Fonseca dos Santos Neto

Co-advisor:

Márcio de Medeiros Ribeiro

MACEIÓ, AUGUST OF 2023

Juliano Rocha Barbosa

Evaluating the Resilience of Cloud NLP Services across Amazon, Microsoft, and Google

Dissertation presented as a partial requirement for obtaining
a Master's degree in the Graduate Program in Informatics at
the Institute of Computing, Federal University of Alagoas.

Advisor:

Baldoino Fonseca dos Santos Neto

Co-advisor:

Márcio de Medeiros Ribeiro

Maceió, August of 2023

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecária Responsável: Livia Silva dos Santos - CRB 1670

B238e Barbosa, Juliano Rocha.

Evaluating the resilience of cloud NLP services across amazon, microsoft, and google /
Juliano Rocha Barbosa. – 2023.

38 f. : il.

Orientador: Balduino Fonseca dos Santos Neto.

Coorientador: Márcio de Medeiros Ribeiro.

Dissertação (Mestrado em Informática) – Universidade Federal de Alagoas. Instituto
de Computação. Programa de Pós-Graduação em Informática. Maceió, 2023.

Bibliografia: f. 36-38.

1. Processamento de Linguagem Natural. 2. Serviço de nuvem. 3. Ruídos (Informática).
I. Título.

CDU: 004.04



UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Informática – PPGI
Instituto de Computação/UFAL
Campus A. C. Simões BR 104-Norte Km 14 BL 12 Tabuleiro do Martins
Maceió/AL - Brasil CEP: 57.072-970 | Telefone: (082) 3214-1401




Folha de Aprovação

JULIANO ROCHA BARBOSA


AVALIANDO A RESILIÊNCIA DE SERVIÇOS NLP NA AMAZON, MICROSOFT E GOOGLE

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas e aprovada em 30 de agosto de 2023.


Banca Examinadora:

Documento assinado digitalmente
 **BALDOINO FONSECA DOS SANTOS NETO**
Data: 17/10/2023 14:23:51-0300
Verifique em <https://validar.iti.gov.br>


Prof. Dr. BALDOINO FONSECA DOS SANTOS NETO
UFAL – Instituto de Computação
Orientador

Documento assinado digitalmente
 **MARCELO COSTA OLIVEIRA**
Data: 19/10/2023 15:49:52-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. MARCELO COSTA OLIVEIRA
UFAL – Instituto de Computação
Examinador Interno

Documento assinado digitalmente
 **MARCIO DE MEDEIROS RIBEIRO**
Data: 19/10/2023 16:24:19-0300
Verifique em <https://validar.iti.gov.br>

MARCIO DE MEDEIROS RIBEIRO
UFAL – Instituto de Computação
Coorientador

Documento assinado digitalmente
 **LEOPOLDO MOTTA TEIXEIRA**
Data: 17/10/2023 14:50:27-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. LEOPOLDO MOTTA TEIXEIRA
UFPE-Universidade Federal de Pernambuco
Examinador Externo

I dedicate this work to all those who cannot start a master's degree due to financial reasons, either because they have more urgent immediate concerns such as being able to feed themselves, or because they are too busy trying to build something for themselves and their family.

Acknowledgments

A passage of direct knowledge, a stimulus at the right moment, or just an energizing conversation. If I have reached where I am, it's due to a long list of people who allowed me to do so. Therefore, I would like to express my gratitude to some individuals.

First, I would like to thank God for the knowledge and experiences granted to me, for the strength to pursue them, and for the people I have met on this journey.

Among these people, I am immensely thankful to my wife Mylena for her patience, encouragement, and for allowing me to dedicate so much time to my academic tasks while we planned our life together. Thank you.

I am grateful to Prof. Dr. Balduino Fonseca for enriching and significantly contributing to my intellectual development. I also want to thank Prof. Dr. Davy de Medeiros Baia for his guidance and support throughout the various stages of completing this work.

Resumo

O Processamento de Linguagem Natural (PLN) revolucionou indústrias, agilizando o atendimento ao cliente por meio de aplicações na área de saúde, finanças, direito, recursos humanos e simplificando tarefas como pesquisa médica, análise financeira e análise de sentimentos. Para evitar os altos custos de construção e manutenção da infraestrutura de PLN, as empresas recorrem aos serviços de PLN em nuvem oferecidos por grandes provedores de nuvem como Amazon, Google e Microsoft. No entanto, há pouco conhecimento sobre o quão resilientes esses serviços são quando sujeitos a ruídos. Este artigo apresenta um estudo que analisa a resiliência dos serviços de PLN em nuvem, avaliando a eficácia dos serviços de análise de sentimentos fornecidos pela Amazon, Google e Microsoft quando submetidos a 12 tipos de ruído, incluindo ruídos sintáticos e semânticos. Os resultados indicam que o Google é o mais resiliente a ruídos sintáticos, e a Microsoft é a mais resiliente a ruídos semânticos. Essas descobertas podem ajudar desenvolvedores e empresas na escolha do provedor de serviços mais adequado e lançar luz sobre a melhoria das técnicas de ponta para serviços de PLN em nuvem eficazes.

Palavras-chave: Processamento de Linguagem Natural, Serviços em Nuvem, Robustez.

Abstract

Natural Language Processing (NLP) has revolutionized industries, streamlining customer service through applications in healthcare, finance, legal, and human resources domains, and simplifying tasks like medical research, financial analysis, and sentiment analysis. To avoid the high costs of building and maintaining NLP infrastructure, companies turn to Cloud NLP services offered by major cloud providers like Amazon, Google, and Microsoft. However, there is little knowledge about how resilient these services are when subjected to noise. This paper presents a study that analyzes the resilience of Cloud NLP services by evaluating the effectiveness of sentiment analysis services provided by Amazon, Google, and Microsoft when subjected to 12 types of noise, including syntactic and semantic noises. The findings indicate that Google is the most resilient to syntactic noises, and Microsoft is the most resilient to semantic noises. These findings may help developers and companies in selecting the most suitable service provider and shed light towards improving state-of-the-art techniques for effective cloud NLP services.

Keywords: Natural Language Processing, Cloud Services, Robustness.

List of Figures

Figure 1 – Word count distribution of sentences in the dataset per class.	18
Figure 2 – Word count distribution of sentences in the dataset.	19

List of Tables

Table 1 – Impact of Keyboard and OCR noises	2
Table 2 – Predictions from cloud-based sentiment analysis services for noise-free sentences	2
Table 3 – Predictions from cloud-based sentiment analysis services for sentences with 10% of OCR noise	3
Table 4 – Noise progression for syntactic noises for sentence “I want to fly to Hawaii”	14
Table 5 – Noise insertion progression for semantic noises for sentence “I want to fly to Hawaii”	15
Table 6 – RQ1 - F-Measure variation according to Noise Level	22
Table 7 – Amazon effectiveness for sentences with 12, 19 and 23 words	27
Table 8 – Google effectiveness for sentences with 12, 19 and 23 words	29
Table 9 – Microsoft effectiveness for sentences with 12, 19 and 23 words	31

Contents

List of Figures	iv
List of Tables	v
Contents	vi
1 Introduction	1
1.1 Context and Problem	1
1.2 Objectives and Methodological Aspects	3
1.3 Contributions	4
1.4 Thesis Structure	4
2 Background	5
2.1 Natural Language Processing	5
2.2 Noise	6
2.3 Cloud Services	6
2.4 Evaluation Metrics	7
3 Related Work	9
4 Study Design	11
4.1 Dataset and Services	11
4.2 Noises	11
4.3 Evaluation Process	13
5 Results and Discussions	21
5.1 RQ1. How resilient are the Cloud NLP services when subjected to noise?	21
5.2 RQ2. Does sentence length influence the resiliency of Cloud NLP services to noise?	26
6 Threats to Validity	34
7 Conclusion	35
Bibliography	36

1 Introduction

In this Chapter, we present a summary of the research, starting with the context and problem, connecting them with the objectives and contributions of this work.

1.1 Context and Problem

Natural Language Processing (NLP) is a branch of ML research and applications that incorporates computer programming that automatically understand and analyze natural language text (JANG et al., 2022). NLP has emerged as a disruptive technology across industries, revolutionizing the way businesses interact with customers and process information (GALBUSERA; CASAROLI; BASSANI, 2019; BAHJA, 2020). NLP technology can leverage the creation of innovative applications; however, it is very costly for companies to build and maintain their own NLP infrastructure. This high cost is due to the need for highly qualified labor to train and implement services of this type, in addition to the high investment in computing power to provide these services (PAJOLA; CONTI, 2021; SHEN; KRIMPEN; SPRUIT, 2019). In this context, major cloud providers like Amazon, Google, and Microsoft have provided a variety of Cloud NLP services (PAJOLA; CONTI, 2021; ARAUJO et al., 2018). These services are a category of cloud-based solutions that allow developers and businesses to integrate powerful NLP techniques into their applications without the need to build and maintain their own NLP infrastructure.

Although Cloud NLP services are a promising way to popularize NLP techniques, developers and businesses may need help selecting the service that best fits their needs. Mainly because these services can be subjected to noises when deployed in real-world situations (ZHANG et al., 2022). By noise, we mean any disturbance to the data that interferes with effective text processing and analysis (SAEED et al., 2021). In Table 1.1, we have two types of noise studied in this work. The first of them we call "Keyboard", this noise occurs due to typing errors when the user mistakenly exchanges a letter in a word for another nearby letter on the keyboard. The second, we call "OCR" and occurs due to failures in the process of scanning documents into digital media, in this process a character may end up being exchanged for a similar one. Typing errors are common on social networks and scanned documents may have smudges or other disturbances that hinder optical character recognition

(OCR).

Table 1 – Impact of Keyboard and OCR noises

Noise	Sentence	Noised sentence
Keyboard	I want to fly to Hawaii	I want to Ely to TWwaii
OCR	I want to fly to Hawaii	1 want tu fly t0 Hawaii

Dealing with noise is critical to preprocessing and data cleaning in NLP pipelines to ensure the best effectiveness of NLP services (SHAROU; LI; SPECIA, 2021). To better illustrate this scenario, in tables 2 and 3, we have the outputs from the sentiment analysis services of three prominent cloud NLP providers. Table 2 contains predictions from Google, Microsoft, and Amazon for three different sentences without noise, and it's worth noting that in almost all scenarios, the providers correctly classified the sentences. On the other hand, in table 3, we have the outputs of the same services for the same sentences, but now 10% of each sentence has been altered with the noise OCR. In this second scenario, it is observed that almost all providers now classify the sentences incorrectly. This shows us how the presence of noise can interfere with text processing and analysis.

Previous studies have investigated the effectiveness of NLP techniques by applying different approaches, such as noise insertion into NLP applications (NÁPLAVA et al., 2021; MORADI; SAMWALD, 2021; BELINKOV; BISK, 2017; RYCHALSKA et al., 2019; NÁPLAVA et al., 2021; MORADI; SAMWALD, 2021; RYCHALSKA et al., 2019; PALLAS; STAUFER; KUHLENKAMP, 2020), and the use of adversarial examples as input to NLP techniques (PAJOLA; CONTI, 2021; BOUCHER et al., 2022). However, there is little knowledge about the resilience of Cloud NLP services when subjected to noise inherent to real-world situations. By resilience, we mean how influential noises are to the effectiveness of Cloud NLP services.

Table 2 – Predictions from cloud-based sentiment analysis services for noise-free sentences

Original sentences	Sentiment	Google	Microsoft	Amazon
@AmericanAir Thanks so much	positive	positive	positive	positive
@AmericanAir Always enjoy my time Now on the plane to DFW	positive	positive	positive	positive
@united go bankrupt again and transfer all of your assets to LUV That would be great	negative	negative	positive	negative

Table 3 – Predictions from cloud-based sentiment analysis services for sentences with 10% of OCR noise

Sentences with 10% of OCR noise	Sentiment	Google	Microsoft	Amazon
@AmericanAir Thanrs 80 much	positive	negative	neutral	neutral
@AmekicanAir A1way8 enjuy my time Now on the p1ane to 0FW	positive	neutral	neutral	neutral
@united go banrrrupt again and tran8fek all 0f yuur a8set8 to LOV That would be gkeat	negative	positive	negative	neutral

1.2 Objectives and Methodological Aspects

The goal of this work was to analyze the resilience of Cloud NLP services by evaluating the influence of 12 types of noise on the effectiveness of these services when subjected to these noises. To achieve this, we divided this work into two research questions:

RQ1. How resilient are the Cloud NLP services when subjected to noise?

This research question analyzes the resilience of Cloud NLP services provided by Google, Microsoft, and Amazon when subjected to 12 types of noise such as typing and spelling errors. As a result of this analysis, we expect to reveal the influence of these types of noise on the effective- ness of the Cloud NLP services analyzed in our study. This way, we expect to help developers select the service providers that best fit their needs

RQ2. Does sentence length influence the resiliency of Cloud NLP services to noise?

As with different types of noise, practitioners and researchers may encounter different sentence lengths depending on the nature and purpose of their work. In this RQ, we want to analyze if the length of sentences impacts the resiliency of the analyzed NLP services when they are subjected to noise. For this, we selected sentences of 3 different lengths and analyzed the influence of 12 types of noise in each group.

To perform our study, we use a dataset containing sentences annotated with positive, negative, or neutral sentiments. Then, we apply the analyzed types of noise to these sentences by varying the noise level. Next, we submit the sentences with noise to the Cloud NLP services analyzed in our study. Finally, we gauge the effectiveness of these services by employing the f-measure as our evaluation metric.

1.3 Contributions

Our results show in which noise scenarios each provider is most resilient. For example, Google is more resilient to syntactic noise such as Keyboard and OCR and Microsoft is more resilient to semantic noise, which uses techniques such as Deep Learning to change sentences. The difference is that while syntactic noise causes a more profound change, semantic noise keeps at least one syntactic structure intact. Regarding sentence length, Amazon has better effectiveness with sentences of 12 and 19, and Google and Microsoft with 12. On the other hand, the effectiveness of Google and Microsoft is lower with sentences of 23 and 19, respectively.

As an implication of the results of our study, we expect to help developers and business to select the service that best fit their needs when using Cloud NLP services. Also, our results shed light towards improving state-of-the-art techniques for effective Cloud NLP services, making these services more reliable.

1.4 Thesis Structure

The remainder of the work is organized as follows:

- Chapter 2, a theoretical framework with the main themes shown in this work will be showed.
- Chapter 3 present and discusses the related work.
- Chapter 4 introduces our research questions and methodological steps.
- Chapter 5 describes the main results and findings identified in our study.
- Chapter 6 describes threats to validity.
- Chapter 7 presents our conclusion which could lead to future contributions.

2 Background

2.1 Natural Language Processing

Natural Language Processing (NLP) is a collection of computational techniques for automatic analysis and representation of human languages (CHOWDHARY; CHOWDHARY, 2020). NLP's data analysis capabilities extract valuable insights from unstructured data, aiding decision-making and research. In healthcare, finance, legal, and human resources domains, NLP simplifies tasks like medical research, financial analysis, legal document processing, and sentiment analysis. Moreover, NLP underpins virtual assistants and smart homes, enriching daily life with voice-controlled automation (ERYIGIT; CELIKKAYA, 2017; GIACHOS et al., 2023).

NLP encompasses a range of tasks, among which we can mention: Text Classification, Named Entity Recognition, Question Answering, Text Summarization, Machine Translation, Sentiment Analysis, and others. Typically, the classical approaches for NLP normally combine syntax analysis and information extraction where documents are represented as vectors, e.g TF-IDF or bag-of-words ones (THO, 2022). During the 20th century, computational resources developed rapidly and, as a result, the Deep Learning (DL) technique proved efficient in solving several computer science problems, including NLP, where it reached notable milestones. One of the first milestones in the area was the introduction of the Word Embedding technique which allows encoding each word in a document into a numerical vector, which is interpreted from the contexts that this word is likely to appear in a training corpus (THO, 2022). This was an important milestone as it made it possible to represent a document as a matrix or a set of vectors that are typically used by Deep Learning architectures such as Convolutional Neural Network (CNN) or Long Short-Term Memory. After that, new advanced text-oriented techniques emerged such as the sequence-to-sequence (seq2seq) architecture. Currently, there are pre-trained language models such as Transformer, BERT or GPT that have significantly improved the performance of NLP tasks and are therefore considered state-of-the-art (THO, 2022).

The first step in the natural language processing pipeline is usually the pre-processing step, in which, in addition to being able to deal with any types of noise or problems in the data, the data can also be transformed to facilitate the functioning of the algorithms. For ex-

ample, the tokenization process can be carried out, where words or terms are exchanged for tokens that represent a category or set of words with the same semantic function. Other techniques used during this process include stemming and lemmatization. After pre-processing, the data modeling stage begins where the chosen techniques are used to generate a model capable of performing the desired tasks (THO, 2022).

The Sentiment analysis task, in particular, involves the analysis and interpretation of opinions, sentiments, and emotions expressed in text or speech. The field of sentiment analysis can be defined as the process of determining the sentiment or emotional tone expressed in a piece of text, whether it is positive, negative, or neutral. The interest in sentiment analysis has been growing in recent years. Commonly, the use of sentiment analysis is related to analyzing public data of products and social media texts (MäNTYLÄ; GRAZIOTIN; KUUTILA, 2018; SCHOUTEN; FRASINCAR, 2016).

2.2 Noise

Noise is a random variance error of a measured variable (KALAPANIDAS et al., 2003). In the context of NLP, noise refers to non-standard textual content (SHAROU; LI; SPECIA, 2021) that causes unnecessary text processing, e.g., inconsistent data, duplication, and wrong writings that can interfere with the effective processing and analysis of the text (SAEED et al., 2021). Noise is common in data sources and applications such as chats, emails, forums, and social media.

2.3 Cloud Services

Cloud services, also known as cloud computing, refer to the delivery of computing resources and services over the internet on demand (ARMBRUST et al., 2010). Rather than bear the high costs and risk of maintaining local servers, cloud services allow users to access and utilize a wide range of resources including processing power, software applications, databases, and machine-learning algorithms from remote servers maintained by cloud service providers (ARMBRUST et al., 2010; GRANA, 2020). With the improvement and popularization of machine-learning algorithms, companies have developed cloud machine-learning services (RIBEIRO; GROLINGER; CAPRETZ, 2015; GOODMAN; XIN, 2020;

PAJOLA; CONTI, 2021). This way, developers and businesses have access to machine learning processes efficiently from anywhere without any concern about implementation details and computing resources (RIBEIRO; GROLINGER; CAPRETZ, 2015).

2.4 Evaluation Metrics

Evaluation metrics are a type of quantitative evaluation, that is, a mechanical way of quantifying the results of a system in numbers (DALIANIS, 2018). Evaluation metrics are useful during the model training and validation process (ERICKSON; KITAMURA, 2021). Commonly, some metric is used as a measure of model training progress. Knowing this, the best metric to be used depends on the domain of the studied problem. For example, in systems where the damage caused by a false negative is greater than a false positive, should be used a metric that penalizes more false negatives.

During the evaluation of a binary classifier, where the objective is to predict whether a sample belongs to one of two possible classes (ERICKSON; KITAMURA, 2021), the results of the classifier are grouped into 4 possible groups:

True positives (TP): These are instances where the model correctly predicts the positive class and the actual ground truth label is also positive.

True negatives (TN): These are instances where the model correctly predicts the negative class and the actual ground truth label is also negative.

False positives (FP): These are instances where the model predicts the positive class, but the actual ground truth label is negative.

False negatives (FN): These are instances where the model predicts the negative class, but the actual ground truth label is positive.

It is common to represent these 4 values in a 2x2 table. This table is known as a *confusion matrix*. The 4 values of a confusion matrix can be used to calculate various metrics such as accuracy, recall, precision, and f-measure (also called the F1-score).

Precision is the fraction of truly positive cases from all cases the model predicted positive and can be calculated using the formula: $\frac{TP}{TP+FP}$ (ERICKSON; KITAMURA, 2021). In turn,

recall is the fraction of positive cases predicted as positive and can be calculated using the formula: $\frac{TP}{TP+FN}$ (ERICKSON; KITAMURA, 2021). F-measure is a measure of the balance between precision and recall and can be calculated using the formula: $\frac{2*Precision*Recall}{Precision+Recall}$. F-measure is commonly used because it provides a single value that combines both precision and recall, allowing for a comprehensive evaluation of the system's performance.

Classification tasks in machine learning involving more than two classes are known by the name of "multi-class classification" (GRANDINI; BAGLI; VISANI, 2020). The sentiment analysis task, when it involves the "negative", "neutral" and "positive" classes, is an example of a multi-class classification task. Most binary classification metrics are easily extended to work with multiclass classifiers, but there are multiple ways to extend a metric as a multiclass (ERICKSON; KITAMURA, 2021).

One way to calculate metrics for multiclass classifiers is to calculate the metric in question for each class against all other classes (GRANDINI; BAGLI; VISANI, 2020). For example, if we were to calculate f-measure for a sentiment analysis model, we would need to calculate f-measure for each of the sentiment classes. In this example, when calculating the metric for the "neutral" class, we would consider the values for the neutral class in the confusion matrix as "positive" values, and all other classes' values as "negative". This method is called "One-vs-Rest" and this way we would have 3 distinct values for f-measure, one for each existing class. There are a few ways to combine results across classes to get a single metric, one of which is called "weighted" where an average of the values is calculated.

3 Related Work

Náplava *et al.* (NÁPLAVA et al., 2021) propose a framework that introduces probabilistic noises based on real-world scenarios. The authors evaluate the resilience of NLP downstream systems when subjected to these noises. The results of this evaluation indicate that probabilistic noises can influence the effectiveness of the systems. Comparing with our study, (NÁPLAVA et al., 2021) focuses on evaluating systems supported by white-box NLP techniques and a specific type of probabilistic noise. By white-box NLP techniques, we mean that the researchers know how the techniques work. Although this does not change how the evaluation is performed, it significantly affects the relevance and applicability of the study. On the other hand, our study analyzes Cloud services supported by black-box NLP techniques, i.e., we do not know how these techniques are implemented by *Google*, *Microsoft*, and *Amazon*. In addition, another difference is that we inserted each noise individually, aiming to better identify the influence of each type of noise on the effectiveness of the services.

In Moradi and Samwald (MORADI; SAMWALD, 2021), the authors investigate the resilience of white-box NLP techniques when subjected to noises based on single changes in characters and words. The results indicate that these noises influence the effectiveness of the models and the models' performance can decrease even when the input contains slight noise. In our work, we use similar perturbation techniques, but the main differences are that in our work we vary the noise intensity and, in our case, we test black-box models.

Belinkov and Bisk (BELINKOV; BISK, 2017) focus on analyzing Neural Machine Translation (NMT) techniques to different types of noise. The authors evaluate the effectiveness of these techniques when subjected to natural (i.e., based on real scenarios) and artificial (i.e., based on existing algorithms) types of noise. The results indicate a decay of the NMT techniques' effectiveness when subjected to texts slightly changed with noise. Our study focuses on sentiment analysis techniques instead of NMT techniques, however, Belinkov and Bisk provided inspiration for the need for analysis aimed at real applications and the use of natural noise in the process.

The work of Rychalska (RYCHALSKA et al., 2019) complement the work of Belinkov and Bisk (BELINKOV; BISK, 2017) beyond the context of character-based machine translation models. The work presents *WildNLP*, a framework for robustness testing to corruption

in machine learning models for NLP. Several forms of text corruption are presented, some of which are also used in this work. The authors also validated the strategy of using adversarial training to increase model robustness. Regarding the way of inserting corruptions in text and evaluating the results, we used a methodology similar to the one presented in this work with the addition of more noise intensities. When compared to our work, the main limitation of this work is that the authors use state-of-the-art models available to the public, whereas in this work the target is ML services which are, in their nature, black-box.

The work of Pallas *et al.* (PALLAS; STAUFER; KUHLENKAMP, 2020) evaluates the effectiveness of Cloud NLP services provided by Google, Microsoft, and IBM in a controlled scenario. The authors analyzed the accuracy of three different NLP tasks: Sentiment Analysis, Named Entity Recognition, and Text Classification while in our work we focus only on the sentiment analysis task. This allows us to deepen the analysis by considering the addition of noise when analyzing the effectiveness of services. Our aspiration is to get closer to the practical application of these Cloud NLP services in real-world scenarios through the introduction of diverse noise types.

Adversarial Machine Learning is a popular area in the field of Natural Language Processing (NLP). Adversarial Machine Learning consists of introducing strategically modified instances into machine learning models in order to deceive the models. In Pajola and Conti (PAJOLA; CONTI, 2021) and Boucher (BOUCHER *et al.*, 2022), the authors focus on applying adversarial machine learning techniques to evaluate the effectiveness of NLP services provided by Amazon, Google, IBM, and Microsoft. The results indicate that these services need sanitization in its inputs. Our work focuses on common scenarios, without assuming malicious behavior on the part of the user, we do this by analyzing a variety of types of noises based on real-world scenarios.

As we can see, previous works have analyzed the effectiveness of NLP techniques using different approaches, such as noise insertion and adversarial examples. Additionally, black box and white box models were explored. However, there is little knowledge about the resilience of Cloud NLP services when subjected to noise inherent to real-world situations. This is important as it helps professionals to choose the service provider that best suits their needs. We hope to contribute to the reduction of this gap.

4 Study Design

In our study, we investigated the effectiveness of sentiment analysis services provided by Google, Microsoft, and Amazon when subjected to syntactic and semantic noise, such as typos and spelling errors. Additionally, we verified the influence of the length of the sentence on the performance of these services when subjected to noise. These points were explored through our research questions defined in section 1.2

4.1 Dataset and Services

We analyze the resilience of the sentiment analysis services provided by *Amazon*, *Google*, and *Microsoft*. We investigate these services because they belong to leading companies in the technology industry. Also, sentiment analysis is one of the most common tasks involving NLP. In particular, these services work by classifying sentences according to their polarity (or sentiment): *positive*, *negative*, and *neutral*.

To perform our analysis, we use the dataset *Twitter US Airline Sentiment*¹. This dataset contains sentences annotated as positive, negative, or neutral according to the feelings of travelers about airlines. For the annotation of sentences in this dataset, the creators of the dataset asked multiple contributors to classify each sentence according to the predominant sentiment in it.

4.2 Noises

We evaluate the resilience of the Cloud NLP services when subjected to 12 types of noises. The following taxonomy was based on the nomenclature given by the *nlpaug* library; however, since this library has flexible noises, some noise names were adapted to better describe the parameters used. This name adaptation occurred with the following noises: RandomCharReplace, CharSwap, WordSplit and WordSwap.

Keyboard: It replaces characters at random based on their proximity on a *QWERTY* keyboard layout. This mimics typos that a user could make while typing a document. For example, the user types “noide” instead of “noise”;

¹ Available at: <https://www.kaggle.com/crowdflower/twitter-airline-sentiment>

OCR: It replaces characters on a document simulating a bad OCR² scan. This means that visually similar characters are more likely to be swapped between themselves. For example, the OCR can recognize “n0ise” instead of “noise”;

RandomCharReplace: It replaces characters at random, for example, substituting “noise” by “n3osn”;

CharSwap: It performs a swap between characters that compose a word (BELINKOV; BISK, 2017). For example, it can swap the characters of the word “noise” to “nosie”;

WordSplit: It splits a word. For example, it can split the word “noise” to “no ise”;

WordSwap: It swaps adjacent words in a sentence, for example, it can swap the sentence “a natural noise” to “a noise natural”;

Antonym: It replaces words with their antonyms based on an antonym dictionary. For example, it can replace the word “many” with “few”;

Synonym: It replaces words with their synonyms based on a synonym dictionary. For example, it can replace the word “angry” with “furious”;

Spelling: It introduces spelling mistakes based on a misspelling dictionary. For example, it can replace the word “acquire” with ‘aquire”;

TfIdfWord: It uses the statistics method TF-IDF³ (Term Frequency-Inverse Document Frequency) to insert or substitute words. For example, it can change the sentence “AmericanAir Thanks so much” to “AmericanAir Thanks so anywhere”.

WordEmbeddings: Given a specific word, this type of noise uses the Word embedding GloVe(PENNINGTON; SOCHER; MANNING, 2014) to identify the most similar words. Then, it replaces the specific word with a similar one based on word embedding. For example, it can replace the word “many” with “more”.

² Optical Character Recognition

³ TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.

ContextualWordEmbs: This type of noise works like WordEmbeddings, but *ContextualWordEmbs* works by feeding surrounding words to BERT(DEVLIN et al., 2018) language model to find out the most suitable word for replacement. For example, it can change the sentence “VirginAmerica on all your flights” to “VirginAmerica on all summer flights”.

We select types of noise that can occur in real-world scenarios, aiming to reveal useful insights for developers when selecting Cloud NLP services that best fit the developers’ needs.

To apply these noises, we use the tool *nlpaug* (MA, 2019). This library stands out due to its numerous available noise insertion algorithms and ample customization capacity; e.g., it is possible to choose parameters such as noise level, noise insertion strategy, and more. Among the available tools, this one proved to be the most prominent in these two essential characteristics for this work. The algorithms available in *nlpaug* simulate behaviors such as typos and OCR errors (Optical Character Recognition) and also apply more robust machine learning algorithms for word replacement.

4.3 Evaluation Process

The code developed in this experiment is available in a Jupyter Notebook on *GitHub*⁴. We evaluate the resilience of the Cloud NLP services by performing the following steps:

Data sampling: Our dataset, *Twitter US Airline Sentiment*, has a total of 14600 instances.

Initially, we extract a sample from the dataset aiming to create a balanced dataset containing the same number of instances classified as positive, negative, and neutral. In our study, we analyze the Cloud NLP services provided by Google, Microsoft, and Amazon, as these services have a significant cost, we consider a low number of instances to perform our analysis. As a result, we opted by creating a new dataset containing a total of 99 instances, 33 instances for each class. Tests were also performed on a smaller number of instances, but no relevant results were obtained.

In RQ2, we repeat the data sampling step 3 times for different sentence lengths. We do this because we are interested in studying the impact caused by variation in sentence

⁴ https://github.com/Juliano-rb/experiments_fault_injection_mlaas

Table 4 – Noise progression for syntactic noises for sentence “I want to fly to Hawaii”

Noise	Noise Level (%)	Example
Keyboard	10	I want to Ely to TWwaii
	20	I want t0 fOy to Hawz98
	30	I waBt t9 f.y ^o jawz&i
	80	j QWHY gK D/% hK TSdqu(
	90	K SSjh 5l r:g %L bxAs*J
OCR	10	1 want tu fly t0 Hawaii
	20	1 want t0 f1y tu Hawaii
	30	1 want tu f1y t0 Hawaii
	80	1 want tu f1y t0 Hawaii
	90	1 want t0 f1y t0 Hawaii
RandomCharReplace	10	F wantYt! fly to Hawaii
	20	IX.ant toefly to H.wavi
	30	I want _o +lS eoxL3waii
	80	.ywhTtZ1rf3A0 K@)bEsgG
	90	UBuH*Z6c4TdLyMaa6fA.Oi4
CharSwap	10	I wnat to fly ot Hawaii
	20	I want to fyl ot Ahwaii
	30	wiantt o fly t oHawaii
	80	Wi antt o flyot Haawii
	90	Ia wnt ot fyl toWhaaii
WordSplit	10	I want to fly to H awaii
	20	I wa nt to fly to H awaii
	30	I w ant to fly to Ha waii
	80	I wa nt to fly to Haw aii
	90	I w ant to fly to Ha waii
WordSwap	10	I want to to fly Hawaii
	20	I fly want to to Hawaii
	30	I want to to Hawaii fly
	80	I want Hawaii to fly to
	90	I want to fly to Hawaii

Table 5 – Noise insertion progression for semantic noises for sentence “I want to fly to Hawaii”

Noise	Noise Level (%)	Example
Antonym	10	I want to fly to Hawaii
	20	I want to fly to Hawaii
	30	I want to fly to Hawaii
	80	I want to fly to Hawaii
	90	I want to fly to Hawaii
Synonym	10	I want to fly to Hawaii
	20	I require to fly to Hawaii
	30	I desire to fly to Hawaii
	80	I desire to fell to Hawaii
	90	I require to vanish to Hawaii
Spelling	10	I want to fliing to Hawaii
	20	I’d wantt to fly to Hawaii
	30	I wen to flight to Hawaii
	80	I’d wang e fly so Hawai
	90	Hi went tj fliing ta Havai
TfIdfWord	10	I want 574 fly to Hawaii
	20	I CLEANED to BHM to Hawaii
	30	I want bewhat fly to resort
	80	I given Haiti CakeNDeath raft humans
	90	I routinely Hawaii MichaelBColeman nozzle profitable
WordEmbeddings	10	I want see fly to Hawaii
	20	I make to goin to Hawaii
	30	I want how when to Hawaii
	80	I must ca make ’ll Hawaii
	90	I some need run into Hawaii
ContextualWordEmbs	10	i want to fly from hawaii
	20	to want to fly for hawaii
	30	i had to fly from hawaii
	80	i made her remain above work
	90	dave found her come somewhere somewhere

length. For this, a preliminary study was necessary in the data sampling to determine which sentence lengths we will be analyzing. This study will be presented later in this section;

Oracle: After creating the balanced dataset, we use the f-measure to evaluate the effectiveness of the Cloud NLP services by using this dataset. The result of this evaluation serves as an oracle to be compared with the effectiveness of the Cloud NLP services when subjected to different levels of noise, as described in the next steps;

Noise generation: At this step, we use the tool *nlpaug* to produce different datasets containing sentences changed according to different levels of noise. The tool *nlpaug* works by splitting the sentence into tokens and then, applying noise to each token. Each token is associated with a word. In the *nlpaug*, there is a parameter called *aug_char_p*, which indicates the percentage of characters in the token that must be changed. With the default *aug_char_p* value, *nlpaug* applies noise to 30% of the characters of each token. For example, consider the sentence “Glorious sunsets mesmerize”, if we apply a keyboard noise, a possible result could be “GloriLjQ s&bcets Nesme5iae”. Notice that, approximately 30% of each word has been changed. In our study, we need to analyze the influence of 12 types of noise applied to an entire sentence rather than each word. To do that, we change the default tokenizer of the *nlpaug* to apply the noise in the entire sentence. As a consequence, the parameter *aug_char_p* indicates the percentage of characters that must be changed in a sentence. For example, applying again the keyboard noise with the described change to the same sentence, a possible result could be “yllrious sunse6d m#Qmdrix”. Notice that, approximately 30% of the entire sentence has been changed.

Tables 4 and 5 demonstrate how noise insertion works with different noise levels for syntactic and semantic noises. Note that in some cases, the noisy sentence is the same as the original, this means that the noise algorithm depends on some replacement dictionary and it was not possible to find words to replace in the dictionary. Furthermore, the algorithms that require the use of specific dictionaries or deep learning models are the following:

WordEmbeddings: To apply the algorithm *WordEmbedding*, we use the model

GloVe (PENNINGTON; SOCHER; MANNING, 2014) trained over 2 billion of tweets;

ContextualWordEmbs: We use the model BERT *bert-base-uncased* (DEVLIN et al., 2018);

Spelling: We use the *English* natural misspelling dictionary provided by (BELINKOV; BISK, 2017);

Synonym: We use the synonym dictionary WordNet (FELLBAUM, 1998);

Antonym: We use the antonym dictionary WordNet (FELLBAUM, 1998).

As a result of this step, we produce nine datasets varying the level of noise from 10% to 90%. Although such a large amount of noise would not be found in real-world scenarios, we chose it for completeness and to gain insights into how providers behave in such extreme situations.

Noise Influence: In this step, we evaluate the effectiveness of the Cloud NLP services by using the nine datasets. For each noise level, we send the sentences of the dataset to the Cloud NLP services. Then, we evaluate the effectiveness of the services by calculating the f-measure. As we are dealing with a multi-class classification problem (we have three sentiment classes: negative, neutral, and positive), when calculating the f-measure, we opted for One-vs-Rest strategy with weighted average. In this method, we first calculate the f-measure value for each class, and, then we calculate a weighted average of the resulting values. As a result, we obtain the values of effectiveness associated with the services provided by Google, Microsoft, and Amazon varying the level of noise from 10% to 90%.

For varying sentence lengths in order to archive the Data sampling requirements of RQ2, we created an auxiliary Jupyter Notebook⁵ where we explored different sampling possibilities. As we are multiplying the amount of data to be processed and therefore the amount of requests to the services being made, we chose 3 different sentence lengths. This helps us not to waste resources. In this way, we aim to represent small, medium, and large sentence sizes.

⁵ https://github.com/Juliano-rb/experiments_fault_injection_mlaas/.../auxiliar.ipynb

To help us choosing the sampling values, we defined three requirements that should be verified in the data sets resulting from the chosen sampling method, namely:

1. The method must result in a balanced data set. Following the sample size defined for the rest of the work, it must be possible to select 33 sentences for each class.
2. The method must rely on some kind of metric (no guessing numbers).
3. It is preferable that the method results in distant groups to increase the perspective of results.

Word count distribution of sentences in the dataset per class

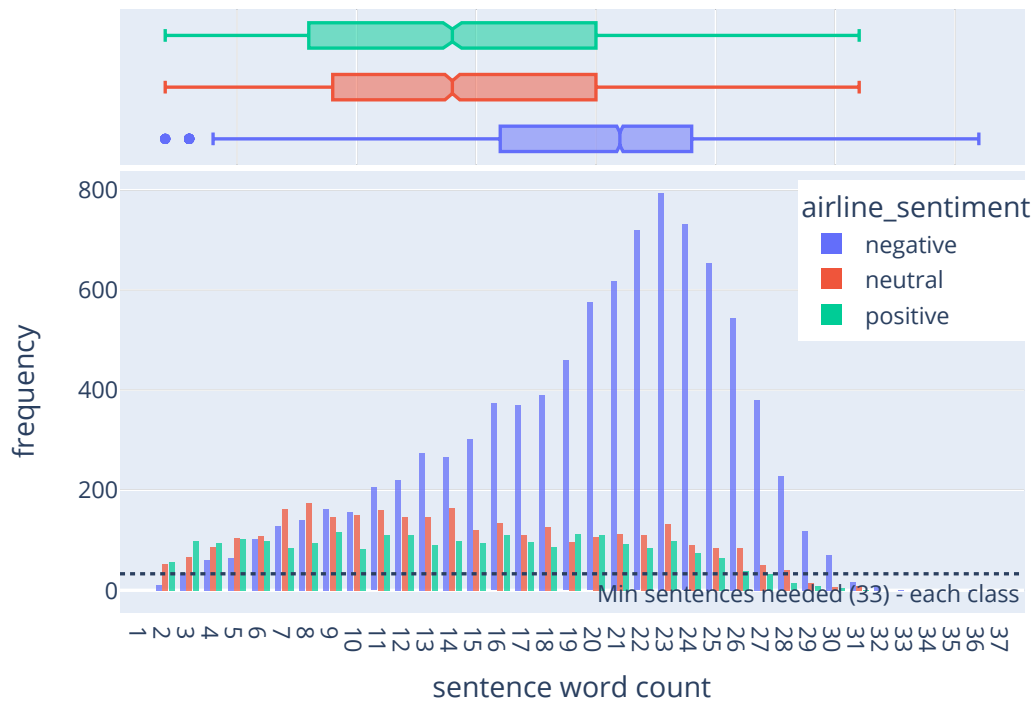


Figure 1 – Word count distribution of sentences in the dataset per class.

To verify the first requirement, we plot Figure 1 which is a histogram of the word count distribution grouped by class for our dataset. In this graph, the x-axis represents the number of words and the y-axis the number of sentences with each number of words. Additionally, in the upper corner, we plot a boxplot of the same data. This visualization of the data shows

us that our initial dataset has more sentences with negative sentiment than neutral or positive. An auxiliary dashed line was also drawn at $y=33$ representing the minimum number of sentences desired in each class. With this, we see that if we select sentences that have a number of words between 3 and 26, the first requirement can be verified.

Word count distribution of sentences in the dataset

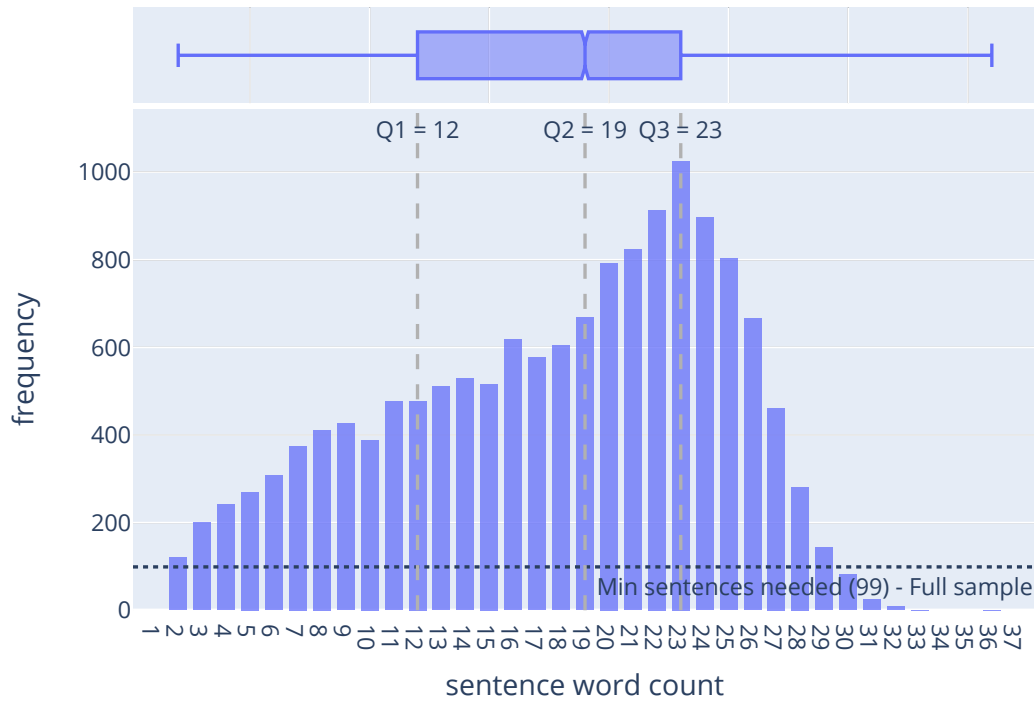


Figure 2 – Word count distribution of sentences in the dataset.

To verify the second requirement, we generated a graph similar to the previous one, but without the distinction by class. This result is illustrated in Figure 2. In this chart we also highlight the threshold values of the three quartiles $Q1$, $Q2$ and $Q3$, which are 12, 19 and 23, respectively. With the quartiles, we have an idea of the sentence length distribution. For example, we know that in $Q2$, that is, sentence length 19, we have the median sentence length of our dataset.

In our analysis, we consider quartile boundaries as group delimiters. In this way, we consider that $Q1$ and $Q2$ represent a good distance from the center of our distribution. Therefore, we chose the $Q1 = 12$, $Q2 = 19$, and $Q3 = 23$ values for our sampling because they repre-

sent our dataset well and have a solid metric for choosing sentence lengths. In our study we opted for the exact values instead of a threshold between the quartiles, this way we increased the difference between the groups, mitigating the effect of the chosen values being close.

5 Results and Discussions

In this section, we present the results of the experiments referring to the research questions defined in the Study Design section. These results are presented in the subsections below.

5.1 RQ1. How resilient are the Cloud NLP services when subjected to noise?

Table 6 presents the effectiveness of the Cloud NLP services when subjected to noise levels from 0% to 90%. The first column describes the noises analyzed in our study, the second column describes the providers of the Cloud NLP services in which we apply the noise and the remaining columns describe the effectiveness of the Cloud NLP services provided by Amazon, Google, and Microsoft. We use a color scale varying from *yellow* to *red* to represent the influence of the noise level on the effectiveness of the Cloud NLP services. The lower the effectiveness, the greater the influence of noise.

We proceed with our analysis by individually examining each noise in table 6. When investigating each noise type, we compared the f-measure values for each provider, highlighting crucial aspects such as identifying the provider with the most significant f-measure decline and ascertaining whether this decline reaches a maximum threshold. We also took note of the absolute numerical range of f-measure variations among the providers to facilitate meaningful comparisons. Throughout this process, we discerned certain noise types that exert minimal impact on the services' f-measures. Additionally, we made relevant comparisons between these noise types using the noise categories we have established in this study: Syntactic and Semantic. Lastly, these two categories were instrumental in determining which type of noise each provider demonstrates greater resilience towards.

Initially, we apply no noise to the Cloud NLP services. Thus, the providers (Amazon, Google, and Microsoft) present effectiveness varying between 0.69 (Google) and 0.76 (Microsoft), regardless of the noise analyzed. Then, we apply noises varying from 10% up to 90%. In such cases, we observe a variation of the effectiveness as follows.

Keyboard. As we increase the level of the noise **Keyboard**, Microsoft presents effectiveness between 0.76 (0% noise level) and 0.17 (50%-90% noise level), decaying up to 0.59

Table 6 – RQ1 - F-Measure variation according to Noise Level

		Noise Level (%)									
		0	10	20	30	40	50	60	70	80	90
Keyboard	Amazon	0.72	0.58	0.35	0.24	0.18	0.21	0.19	0.17	0.16	0.17
	Google	0.69	0.62	0.46	0.41	0.33	0.36	0.28	0.23	0.28	0.34
	Microsoft	0.76	0.66	0.34	0.18	0.19	0.17	0.17	0.17	0.17	0.17
OCR	Amazon	0.72	0.47	0.27	0.21	0.22	0.19	0.20	0.23	0.22	0.24
	Google	0.69	0.59	0.45	0.44	0.41	0.40	0.39	0.40	0.45	0.44
	Microsoft	0.76	0.59	0.32	0.21	0.17	0.19	0.19	0.19	0.19	0.17
RandomCharReplace	Amazon	0.72	0.53	0.30	0.24	0.22	0.17	0.17	0.17	0.17	0.17
	Google	0.69	0.54	0.52	0.33	0.22	0.32	0.26	0.21	0.26	0.21
	Microsoft	0.76	0.62	0.44	0.34	0.20	0.19	0.16	0.17	0.19	0.16
CharSwap	Amazon	0.72	0.61	0.54	0.44	0.40	0.30	0.37	0.29	0.24	0.29
	Google	0.69	0.64	0.46	0.50	0.49	0.37	0.40	0.48	0.41	0.34
	Microsoft	0.76	0.74	0.57	0.42	0.41	0.26	0.27	0.16	0.18	0.16
WordSwap	Amazon	0.72	0.71	0.75	0.71	0.70	0.66	0.72	0.71	0.68	0.70
	Google	0.69	0.62	0.71	0.62	0.66	0.60	0.64	0.59	0.66	0.62
	Microsoft	0.76	0.74	0.77	0.72	0.74	0.70	0.72	0.69	0.69	0.75
WordSplit	Amazon	0.72	0.63	0.63	0.57	0.48	0.43	0.39	0.36	0.34	0.34
	Google	0.69	0.63	0.58	0.58	0.51	0.50	0.44	0.39	0.38	0.38
	Microsoft	0.76	0.68	0.64	0.65	0.49	0.40	0.35	0.30	0.30	0.30
Antonym	Amazon	0.72	0.67	0.59	0.58	0.58	0.57	0.59	0.58	0.61	0.59
	Google	0.69	0.55	0.50	0.48	0.48	0.44	0.45	0.47	0.44	0.46
	Microsoft	0.76	0.68	0.61	0.60	0.63	0.65	0.62	0.64	0.63	0.61
Synonym	Amazon	0.72	0.74	0.71	0.73	0.69	0.67	0.64	0.63	0.57	0.63
	Google	0.69	0.66	0.66	0.68	0.68	0.69	0.61	0.63	0.61	0.61
	Microsoft	0.76	0.72	0.75	0.70	0.70	0.62	0.68	0.64	0.67	0.67
Spelling	Amazon	0.72	0.66	0.70	0.58	0.61	0.54	0.48	0.44	0.44	0.49
	Google	0.69	0.67	0.62	0.65	0.68	0.57	0.60	0.48	0.47	0.53
	Microsoft	0.76	0.76	0.69	0.66	0.67	0.59	0.56	0.53	0.55	0.50
TfIdfWord	Amazon	0.72	0.67	0.59	0.53	0.48	0.47	0.37	0.38	0.22	0.28
	Google	0.69	0.58	0.58	0.60	0.55	0.51	0.53	0.38	0.28	0.39
	Microsoft	0.76	0.70	0.55	0.53	0.58	0.48	0.46	0.41	0.35	0.34
WordEmbeddings	Amazon	0.72	0.69	0.64	0.61	0.48	0.61	0.53	0.43	0.48	0.46
	Google	0.69	0.59	0.56	0.52	0.55	0.59	0.45	0.53	0.49	0.39
	Microsoft	0.76	0.69	0.72	0.61	0.63	0.66	0.56	0.57	0.54	0.57
ContextualWordEmbs	Amazon	0.72	0.66	0.59	0.66	0.54	0.48	0.51	0.36	0.38	0.45
	Google	0.69	0.63	0.67	0.62	0.55	0.55	0.51	0.42	0.42	0.45
	Microsoft	0.76	0.73	0.66	0.64	0.54	0.61	0.59	0.38	0.50	0.53

in effectiveness. Amazon presents a decay of up to 0.56 in effectiveness, varying the effectiveness between 0.72 and 0.16. Google presents effectiveness between 0.69 and 0.23, decaying up to 0.46 in effectiveness. Notice also that, from 20% noise level, Google reaches effectiveness greater than the other providers. *These findings indicate that the **Keyboard** may lead the Cloud NLP services to decrease its effectiveness by more than 50%, and Google is the most resilient provider to this noise.*

OCR. Similar to **Keyboard**, Microsoft presents the highest decay and Google presents the lowest decay when subjected to the **OCR**. Microsoft presents a decay of 0.59, reaching effectiveness between 0.76 and 0.17, and Google presents a decay of 0.30, reaching effectiveness between 0.69 and 0.39. Amazon presents effectiveness between 0.72 and 0.19, decaying up to 0.53 in effectiveness. From a 10% noise level, Google presents effectiveness greater than the other providers. *These findings indicate that the **OCR** may lead the Cloud*

NLP services to decrease its effectiveness by more than 50%, and Google is the most resilient provider to this noise.

RandomCharReplace. Just as **OCR** and **Keyboard**, Microsoft presents the highest decay and Google presents the lowest decay when subjected to the **RandomCharReplace**. Microsoft presents a decay of 0.60, reaching effectiveness between 0.76 and 0.16, and Google presents a decay of 0.48, reaching effectiveness between 0.69 and 0.21. Amazon presents effectiveness between 0.72 and 0.17, decaying up to 0.55 in effectiveness. From a 10% noise level, Google presents effectiveness equal to or greater than the other providers. *These findings indicate that the **RandomCharReplace** may lead the Cloud NLP services to decrease its effectiveness by up to 60%, and Google is the most resilient provider to this noise.*

CharSwap. Similar to **OCR**, **Keyboard**, and **RandomCharReplace**, Microsoft presents the highest decay and Google presents the lowest decay when subjected to the **CharSwap**. Microsoft presents a decay of 0.60, reaching effectiveness between 0.76 and 0.16, and Google presents a decay of 0.35, reaching effectiveness between 0.69 and 0.34. Amazon presents effectiveness between 0.72 and 0.24, decaying up to 0.48 in effectiveness. From a 30% noise level, Google presents effectiveness greater than the other providers. *These findings indicate that the **CharSwap** may lead the Cloud NLP services to decrease its effectiveness by up to 60%, and Google is the most resilient provider to this noise.*

WordSwap. Different from the previous noises, the **WordSwap** does not present a relevant influence on the effectiveness of the providers. This noise leads the providers to decrease its effectiveness to a maximum of 0.1. For instance, Google presents effectiveness varying between 0.69 and 0.59. Amazon presents a decay of 0.06, reaching effectiveness between 0.72 and 0.66. Microsoft varies the effectiveness between 0.76 and 0.69, indicating the decay of 0.07 in effectiveness. *These findings indicate the providers are resilient to the **WordSwap**.*

WordSplit. Just as **CharSwap**, **OCR**, **Keyboard**, and **RandomCharReplace**, Microsoft presents the highest decay, and Google presents the lowest decay when subjected to the **WordSplit**. Microsoft presents a decay of 0.46, reaching effectiveness between 0.76 and 0.30, and Google presents a decay of 0.31, reaching effectiveness between 0.69 and 0.38. Amazon presents effectiveness between 0.72 and 0.34, decaying up to 0.38 in effectiveness. From a 40% noise level, Google presents effectiveness greater than the other providers. *These findings indicate that the **WordSplit** may lead the Cloud NLP services to*

decrease its effectiveness by up to 46%, and Google is the most resilient provider to this noise.

So far, we have analyzed syntactic noises' influence on the providers' effectiveness. By syntactic noises, we mean noises based on syntactic changes in the sentence, such as **Keyboard**, **OCR**, **RandomCharReplace**, **CharSwap**, **WordSwap**, and **WordSplit**. For syntactic noises, we observe that most of these types of noise influence the providers' effectiveness. Also, Google is the most resilient provider, and Microsoft is the provider that most suffer from these types of noises. Now, we analyze the influence of semantic noises on the effectiveness of the providers. By semantic noises, we mean noises based on semantic changes in the sentence, such as **Antonym**, **Synonym**, **Spelling**, **TfIdfWord**, **WordEmbeddings**, and **ContextualWordEmbs**, as described in Section 4.2.

Antonym. Different from the syntactic noises, Google presents the highest decay. Google presents a decay of 0.35, reaching effectiveness between 0.69 and 0.44. Amazon and Microsoft present the lowest decay when subjected to the **Antonym**. Amazon presents a decay of 0.15, reaching an effectiveness between 0.72 and 0.57. Microsoft presents a decay slightly greater than Amazon, varying between 0.76 and 0.60. Microsoft presents effectiveness greater than the other providers in all the noise levels analyzed. *These findings indicate that the **Antonym** may lead the NLP services to decrease its effectiveness by up to 35%, and Amazon and Microsoft are more resilient than Google to this noise.*

Synonym. **Synonym** does not have a relevant influence on the effectiveness of the providers, leading to a decay up to 0.17. Amazon presents a decay of only 0.17, reaching effectiveness between 0.74 and 0.57, and Google presents a decay of 0.08, reaching effectiveness between 0.69 and 0.61. Microsoft presents effectiveness between 0.76 and 0.62, decaying up to 0.14 in effectiveness. Although Microsoft presents a decay of 0.14, it has an effectiveness greater than the other providers in seven levels of noise. *These findings indicate that **Synonym** has a small influence on the effectiveness of the providers, and the providers are resilient to this noise.*

Spelling. This noise leads the providers to decay in effectiveness slightly close to each other. Amazon presents the highest decay and Google presents the lowest decay. Amazon presents a decay of 0.28, reaching effectiveness between 0.72 and 0.44, and Google presents a decay of 0.22, reaching effectiveness between 0.69 and 0.47. Microsoft presents effective-

ness between 0.76 and 0.50, decaying up to 0.26 in effectiveness. Although the providers present a decay close to each other, Microsoft presents effectiveness greater than the other providers in six out of 10 levels of noise analyzed. *These findings indicate that the **Spelling** has an influence on the effectiveness of the providers, and Microsoft is the most resilient provider.*

TfIdfWord. When subjected to the **TfIdfWord**, Amazon presents the highest decay. Its effectiveness varies between 0.72 and 0.22. Microsoft and Google present a decay of 0.41 and 0.42, respectively. While Microsoft presents effectiveness between 0.76 and 0.34, Google presents effectiveness between 0.69 and 0.28. Microsoft presents effectiveness greater than the other providers in half of the levels of noise analyzed. *These findings indicate that the **TfIdfWord** has a relevant influence on the effectiveness of the providers, and Microsoft and Google are the most resilient providers.*

WordEmbeddings. **WordEmbeddings** has a small influence on the effectiveness of the providers. Amazon presents a decrease from 0.72 to 0.43, indicating a decay of 0.29 in effectiveness. Followed by Microsoft with a decay of 0.27, varying the effectiveness from 0.69 to 0.42. Google presents the smallest decaying, reaching effectiveness between 0.69 and 0.54. Microsoft presents effectiveness equal to or greater than the other providers. *These findings indicate that **WordEmbeddings** has a small influence on providers, and Microsoft is the most resilient one.*

ContextualWordEmbs. **ContextualWordEmbs** has a small influence on the effectiveness of the providers. Microsoft presents a decrease from 0.72 to 0.38, indicating a decay of 0.38 in effectiveness. Followed by Amazon with a decay of 0.36, reaching effectiveness from 0.72 to 0.36. Google presents the smallest decay, reaching effectiveness between 0.69 and 0.42. Microsoft presents effectiveness greater than the other providers in six of the noise levels analyzed. *These findings indicate that **WordEmbeddings** has a significant influence on providers, and Microsoft is the most resilient one.*

We observe that most of the semantic noises influence the effectiveness of the providers and Microsoft is the most resilient provider when subjected to these types of noise.

Summary: The results indicate that most of the syntactic and semantic noises influence the effectiveness of the providers. While Google is the most resilient when subjected to

syntactic noises, Microsoft is the most resilient when subjected to semantic noises.

5.2 RQ2. Does sentence length influence the resiliency of Cloud NLP services to noise?

Tables 7-9 present the effectiveness of the MLaaS from Amazon, Google, and Microsoft for sentences with word counts of 12, 19, and 23 for each noise algorithm for all noise levels. In each table, the first column describes the noises analyzed in our study, the second column describes the sentence lengths analyzed, and the remaining columns describe the effectiveness of the Cloud NLP services to the different sentence lengths for each noise. We use a color scale varying from yellow to red to represent the influence of the noise level on the effectiveness of the Cloud NLP services. The lower the effectiveness, the greater the influence of noise.

To conduct our analysis, we examined each provider individually by scrutinizing tables 7-9. Within each table, we assessed the provider's performance for various noise levels across different sentence lengths. Specifically, we determined which sentence size consistently yielded the highest f-measure for each noise level. This allowed us to ascertain the optimal sentence length for each provider under different noise conditions. Subsequently, by aggregating data on each provider's performance across various noise types, we gained a comprehensive understanding of their overall preferred sentence length. Likewise, this approach enabled us to identify the least favorable sentence length for providers by following the same procedure as outlined above.

Table 7 presents the effectiveness of the ML services from Amazon for sentences with word counts of 12, 19, and 23 for each noise algorithm for all noise levels. For the no-noise scenario, the Amazon provider's f-measure for sentence lengths 12, 19 and 23 are 0.71, 0.70, and 0.70. Next, we will perform a noise-to-noise analysis by performing observations.

Keyboard. For **Keyboard** noise, no one sentence length stands out. We observed that at the 20% and 40% levels, sentence length 23 is the worst f-measure. However, this sentence length has the best f-measure at the intermediate value of 30%. From 60% noise, it seems that the f-measure stabilizes.

OCR. Considering the **OCR** noise, we notice that from a noise level of 10%, the sentence length where the provider does the best is 19 and the worst is 23. Note that for all noise levels,

Table 7 – Amazon effectiveness for sentences with 12, 19 and 23 words

		Noise Level (%)									
		0	10	20	30	40	50	60	70	80	90
Keyboard	12	0,71	0,47	0,45	0,24	0,25	0,23	0,19	0,17	0,16	0,16
	19	0,70	0,59	0,45	0,21	0,26	0,19	0,17	0,17	0,16	0,17
	23	0,70	0,58	0,40	0,27	0,20	0,19	0,18	0,16	0,19	0,17
OCR	12	0,71	0,58	0,35	0,26	0,24	0,28	0,26	0,28	0,26	0,25
	19	0,70	0,60	0,36	0,34	0,31	0,29	0,34	0,30	0,35	0,34
	23	0,70	0,54	0,30	0,20	0,22	0,19	0,24	0,23	0,21	0,23
RandomCharReplace	12	0,71	0,54	0,42	0,27	0,18	0,17	0,17	0,17	0,17	0,17
	19	0,70	0,64	0,37	0,26	0,24	0,18	0,17	0,17	0,17	0,17
	23	0,70	0,56	0,31	0,26	0,23	0,18	0,19	0,17	0,17	0,17
CharSwap	12	0,71	0,59	0,52	0,48	0,34	0,48	0,34	0,27	0,33	0,20
	19	0,70	0,63	0,61	0,55	0,44	0,44	0,31	0,22	0,28	0,32
	23	0,70	0,61	0,56	0,44	0,42	0,33	0,34	0,34	0,26	0,32
WordSwap	12	0,71	0,68	0,67	0,69	0,68	0,67	0,71	0,70	0,68	0,71
	19	0,70	0,67	0,65	0,66	0,66	0,67	0,64	0,64	0,63	0,63
	23	0,70	0,67	0,71	0,69	0,67	0,69	0,68	0,66	0,68	0,67
WordSplit	12	0,71	0,62	0,57	0,55	0,46	0,43	0,47	0,39	0,41	0,44
	19	0,70	0,68	0,68	0,52	0,42	0,40	0,38	0,39	0,41	0,36
	23	0,70	0,66	0,58	0,52	0,48	0,39	0,35	0,36	0,35	0,36
Antonym	12	0,71	0,60	0,58	0,60	0,59	0,58	0,61	0,55	0,58	0,59
	19	0,70	0,53	0,52	0,50	0,51	0,56	0,51	0,51	0,49	0,52
	23	0,70	0,54	0,50	0,52	0,52	0,54	0,53	0,56	0,55	0,54
Synonym	12	0,71	0,68	0,67	0,70	0,68	0,60	0,61	0,61	0,58	0,59
	19	0,70	0,67	0,63	0,68	0,64	0,68	0,63	0,64	0,55	0,58
	23	0,70	0,64	0,69	0,67	0,66	0,68	0,55	0,64	0,62	0,57
Spelling	12	0,71	0,69	0,64	0,56	0,61	0,50	0,58	0,51	0,53	0,48
	19	0,70	0,61	0,58	0,62	0,51	0,51	0,51	0,39	0,48	0,41
	23	0,70	0,64	0,59	0,62	0,60	0,53	0,48	0,48	0,53	0,46
TfIdfWord	12	0,71	0,63	0,57	0,49	0,44	0,35	0,35	0,23	0,25	0,31
	19	0,70	0,69	0,53	0,56	0,51	0,46	0,29	0,40	0,25	0,32
	23	0,70	0,64	0,58	0,50	0,47	0,37	0,35	0,19	0,24	0,22
WordEmbeddings	12	0,71	0,66	0,65	0,60	0,60	0,53	0,58	0,50	0,49	0,47
	19	0,70	0,62	0,59	0,60	0,59	0,50	0,54	0,41	0,51	0,46
	23	0,70	0,71	0,62	0,61	0,56	0,53	0,54	0,54	0,44	0,46
ContextualWordEmbs	12	0,71	0,66	0,56	0,42	0,50	0,40	0,41	0,39	0,38	0,37
	19	0,70	0,65	0,65	0,56	0,58	0,52	0,47	0,51	0,52	0,47
	23	0,70	0,63	0,54	0,61	0,53	0,47	0,50	0,48	0,32	0,34

the f-measure values for length 19 are the best and for length 23 we have the worst values. We can say that, for scenarios where OCR noise is present, the Amazon provider is better with medium sentences and worse with larger sentences.

RandomCharReplace. Regarding **RandomCharReplace** noise, the sentence length where the provider performed better most times was 12. Likewise, the sentence length where the provider performed worst was 23. It is also noticed that from 50% noise there is little change in f-measure values.

CharSwap. In CharSwap noise, up to 40% noise, the Amazon provider is better in sentences of length 19 and worse in sentences of length 12. After that, the results vary a lot without stabilization.

WordSwap. In WordSwap noise, there is little f-measure variation. Within the perceived variations, we see that for this noise, the provider is better with sentence length 12 and worse

with sentence length 19. This can be seen because the sentence length with the smallest f-measure values is 19 and the one with the largest is 12.

WordSplit. Observing the **WordSplit** noise, we see that at initial noise levels, the provider seems to be best on sentences of length 19 and bad on sentences of length 12. From 50% noise, length 23 is the worst and length 12 does best only occasionally.

Antonym. Considering **Antonym** noise, the provider is best for sentence length 12, as at almost all noise levels the f-measure is the largest at that sentence length. Similarly, the provider does worse on sentences of length 19 as it has more points where the f-measure is lower.

Synonym. Similarly to **Antonym**, in **Synonym** noise, the provider is best for sentence length 12, as at almost all noise levels the f-measure is the largest. The other sentence lengths seem to perform similarly.

Spelling. Comparing to **Synonym**, for **Spelling** noise, the provider is also best for sentence length 12. But we can see that the provider does worse with sentences of length 19.

TfIdWord. In **TfIdWord** noise, the provider does best at sentence length 19, as at various noise levels the f-measure is the largest. On the other hand, the provider does worse at length 12. After noise level 50% there seems to be greater instability in the results.

WordEmbeddings. Regarding the **WordEmbeddings** noise, the provider does best at sentence length 12, where the provider does best at most noise levels. Likewise, the worst sentence length for the provider is 19 as this is where we observe the lowest f-measures with few high values.

ContextualWordEmbs. For **ContextualWordEmbs** noise, we see that the provider is best for sentence length 19, as at most noise levels the f-measure is the best. On the other hand, the provider does worse at length 12. Although at noise levels 0% and 10% of sentence length 12 the provider does well, at most subsequent levels the provider does poorly.

Considering our analysis in the previous paragraphs, we can see that the provider Amazon does better with sentence lengths 12 and 19. We arrived at this conclusion by analyzing the scenarios where the provider was described as performing better considering sentence length.

Table 8 present the effectiveness of the ML services from Google for sentences with word counts of 12, 19 and 23 for each noise algorithm for all noise levels. For the no-noise

Table 8 – Google effectiveness for sentences with 12, 19 and 23 words

		Noise Level (%)									
		0	10	20	30	40	50	60	70	80	90
Keyboard	12	0,68	0,51	0,53	0,39	0,34	0,40	0,37	0,33	0,33	0,31
	19	0,73	0,59	0,49	0,50	0,44	0,40	0,32	0,37	0,31	0,33
	23	0,63	0,49	0,50	0,36	0,34	0,30	0,46	0,34	0,31	0,29
OCR	12	0,68	0,59	0,60	0,49	0,46	0,47	0,48	0,48	0,47	0,45
	19	0,73	0,63	0,51	0,41	0,41	0,42	0,39	0,40	0,41	0,42
	23	0,63	0,59	0,49	0,37	0,41	0,39	0,39	0,36	0,39	0,36
RandomCharReplace	12	0,68	0,53	0,43	0,39	0,27	0,31	0,26	0,30	0,32	0,21
	19	0,73	0,62	0,58	0,43	0,43	0,32	0,27	0,24	0,25	0,25
	23	0,63	0,57	0,32	0,30	0,35	0,41	0,24	0,30	0,27	0,24
CharSwap	12	0,68	0,69	0,56	0,60	0,54	0,37	0,51	0,38	0,34	0,40
	19	0,73	0,61	0,59	0,54	0,57	0,48	0,45	0,51	0,48	0,42
	23	0,63	0,42	0,52	0,53	0,45	0,45	0,45	0,38	0,39	0,35
WordSwap	12	0,68	0,65	0,63	0,66	0,62	0,65	0,63	0,62	0,59	0,64
	19	0,73	0,61	0,61	0,68	0,64	0,56	0,70	0,55	0,60	0,63
	23	0,63	0,58	0,59	0,62	0,65	0,49	0,59	0,51	0,54	0,61
WordSplit	12	0,68	0,55	0,57	0,54	0,62	0,49	0,51	0,45	0,53	0,54
	19	0,73	0,61	0,71	0,53	0,51	0,47	0,50	0,49	0,51	0,42
	23	0,63	0,61	0,61	0,60	0,54	0,52	0,50	0,38	0,47	0,52
Antonym	12	0,68	0,56	0,58	0,57	0,56	0,58	0,57	0,57	0,57	0,58
	19	0,73	0,53	0,47	0,47	0,47	0,50	0,49	0,50	0,50	0,47
	23	0,63	0,49	0,49	0,54	0,50	0,51	0,53	0,53	0,48	0,51
Synonym	12	0,68	0,68	0,65	0,66	0,67	0,63	0,60	0,61	0,64	0,63
	19	0,73	0,61	0,65	0,63	0,61	0,59	0,54	0,50	0,54	0,53
	23	0,63	0,61	0,62	0,53	0,57	0,60	0,58	0,52	0,52	0,54
Spelling	12	0,68	0,69	0,65	0,68	0,65	0,58	0,49	0,57	0,51	0,53
	19	0,73	0,68	0,63	0,67	0,60	0,61	0,58	0,53	0,58	0,50
	23	0,63	0,63	0,54	0,59	0,55	0,55	0,55	0,46	0,43	0,50
TfidfWord	12	0,68	0,64	0,62	0,52	0,58	0,45	0,38	0,38	0,50	0,42
	19	0,73	0,66	0,66	0,54	0,52	0,49	0,41	0,40	0,35	0,41
	23	0,63	0,55	0,55	0,52	0,51	0,40	0,39	0,36	0,34	0,41
WordEmbeddings	12	0,68	0,66	0,66	0,63	0,57	0,60	0,52	0,45	0,49	0,51
	19	0,73	0,63	0,52	0,64	0,48	0,52	0,43	0,45	0,44	0,39
	23	0,63	0,64	0,57	0,56	0,51	0,48	0,45	0,44	0,43	0,48
ContextualWordEmbs	12	0,68	0,66	0,60	0,61	0,50	0,48	0,51	0,48	0,61	0,48
	19	0,73	0,67	0,58	0,52	0,42	0,53	0,47	0,43	0,52	0,53
	23	0,63	0,61	0,49	0,49	0,46	0,52	0,41	0,43	0,35	0,33

scenario, we have that the Google provider's f-measure for sentence lengths 12, 19 and 23 are, respectively, 0.68, 0.73 and 0.63. Next, we will perform a noise-to-noise analysis by performing observations.

Keyboard. Regarding the **Keyboard** noise, despite observing a lot of variation in the results, the Google provider does better with length 19 and has worse results with length 23.

OCR. In **OCR** noise, the provider is better for sentences length 12, because in almost all noise levels the f-measure is the biggest in this sentence length, this happens mainly from 20% noise. Similarly, the provider does worse on sentences of length 23 as it has more points where the f-measure is lower.

RandomCharReplace. For **RandomCharReplace** noise, the sentence length where the provider performed better most times was 19. In other sentence lengths, the provider acts similarly, being worse at initial noise levels of length 23. After 60% the results vary quite.

CharSwap. As for **CharSwap** noise, the sentence length where the provider performed better most times was 19. Likewise, the provider performs worse in sentences of length 23 as it has more points where the f-measure is lower.

WordSwap. Interestingly, in **WordSwap** noise, in general, there is little f-measure variation. Within the perceived variations, we see that for this noise, the provider is slightly better for sentence length 12 and worse for sentence length 23. This can be seen because the sentence length with the smallest f-measure values is 23 and the one with the largest is 12.

WordSplit. Observing the **WordSplit** noise, in general, there is little f-measure variation. Within the perceived variations, we see that for this noise, the provider is slightly better for the sentence length 12. In the other sentence lengths the provider has similar performance.

Antonym. Considering **Antonym** noise, the provider is best for sentence length 12, as at almost all noise levels the f-measure is best at that sentence length. Also, the provider does worse on sentences of length 19 as it has more points where the f-measure is lower.

Synonym. Similarly to **Antonym**, with the **Synonym** noise, the provider is best for sentence length 12, as at almost all noise levels the f-measure is the largest. Furthermore, the provider does worse on sentences of length 23 at initial noise levels and on sentences of length 19 at final noise levels.

Spelling. Regarding the **Spelling** noise, the provider is better for sentence lengths 12 and 19, because in both sentence lengths the provider had high f-measure values. Also, the provider does worse on sentences of length 23.

TfIdWord. In **TfIdWord** noise, the provider does best at sentence length 19, as at various noise levels the f-measure is the largest. On the other hand, the provider does worse at length 23.

WordEmbeddings. Similarly to **TfIdWord**, the provider is also the worst at length 23 with the **TfIdWord** noise, but for this noise the best sentence length is 12.

ContextualWordEmbs. Considering **ContextualWordEmbs** noise, the provider is best for sentence length 12, as at most noise levels f-measure is best. On the other hand, the provider does worse at length 23.

Considering the previously scenarios, we can see that the Google provider does better with sentence length 12, although it is also good with sentence length 19. We arrived at this

conclusion by analyzing the scenarios where the provider was described as performing better considering sentence length. On the other hand, the worst sentence length for Google was 23. We can say this because we had a high occurrence of noise where the length 23 was considered the worst.

Table 9 – Microsoft effectiveness for sentences with 12, 19 and 23 words

		Noise Level (%)									
		0	10	20	30	40	50	60	70	80	90
Keyboard	12	0,72	0,56	0,42	0,20	0,17	0,19	0,17	0,17	0,17	0,17
	19	0,56	0,53	0,46	0,23	0,22	0,21	0,19	0,21	0,19	0,21
	23	0,51	0,46	0,39	0,27	0,18	0,17	0,17	0,16	0,17	0,17
OCR	12	0,72	0,61	0,36	0,23	0,23	0,21	0,23	0,25	0,26	0,25
	19	0,56	0,55	0,35	0,29	0,28	0,29	0,29	0,31	0,30	0,31
	23	0,51	0,50	0,31	0,26	0,28	0,30	0,26	0,29	0,25	0,29
RandomCharReplace	12	0,72	0,57	0,41	0,29	0,18	0,17	0,23	0,17	0,17	0,17
	19	0,56	0,52	0,44	0,28	0,28	0,16	0,17	0,16	0,16	0,17
	23	0,51	0,50	0,39	0,22	0,21	0,17	0,17	0,17	0,17	0,17
CharSwap	12	0,72	0,69	0,53	0,50	0,35	0,39	0,28	0,25	0,24	0,21
	19	0,56	0,51	0,51	0,52	0,50	0,35	0,22	0,24	0,25	0,27
	23	0,51	0,52	0,48	0,44	0,39	0,30	0,32	0,24	0,26	0,25
WordSwap	12	0,72	0,71	0,70	0,70	0,70	0,69	0,73	0,71	0,71	0,74
	19	0,56	0,57	0,51	0,53	0,52	0,51	0,56	0,55	0,56	0,48
	23	0,51	0,50	0,54	0,52	0,53	0,56	0,56	0,53	0,56	0,47
WordSplit	12	0,72	0,64	0,58	0,48	0,44	0,34	0,33	0,33	0,30	0,43
	19	0,56	0,54	0,56	0,50	0,44	0,38	0,35	0,33	0,37	0,32
	23	0,51	0,58	0,51	0,50	0,52	0,35	0,33	0,42	0,43	0,37
Antonym	12	0,72	0,60	0,57	0,62	0,58	0,62	0,57	0,58	0,56	0,59
	19	0,56	0,39	0,36	0,36	0,35	0,34	0,38	0,34	0,35	0,35
	23	0,51	0,41	0,43	0,47	0,44	0,40	0,45	0,48	0,44	0,42
Synonym	12	0,72	0,72	0,69	0,65	0,74	0,69	0,63	0,62	0,60	0,68
	19	0,56	0,44	0,52	0,49	0,51	0,53	0,45	0,50	0,37	0,48
	23	0,51	0,54	0,53	0,53	0,54	0,54	0,46	0,58	0,49	0,51
Spelling	12	0,72	0,71	0,68	0,70	0,61	0,52	0,52	0,58	0,58	0,59
	19	0,56	0,55	0,57	0,50	0,49	0,43	0,49	0,37	0,42	0,45
	23	0,51	0,55	0,53	0,56	0,56	0,55	0,52	0,58	0,54	0,48
TfidfWord	12	0,72	0,65	0,64	0,48	0,46	0,40	0,39	0,39	0,32	0,23
	19	0,56	0,59	0,43	0,38	0,43	0,46	0,36	0,40	0,42	0,40
	23	0,51	0,53	0,53	0,43	0,42	0,41	0,40	0,39	0,34	0,30
WordEmbeddings	12	0,72	0,67	0,64	0,61	0,61	0,57	0,52	0,44	0,47	0,54
	19	0,56	0,54	0,47	0,50	0,42	0,55	0,38	0,36	0,44	0,46
	23	0,51	0,51	0,53	0,53	0,36	0,46	0,46	0,43	0,49	0,47
ContextualWordEmbs	12	0,72	0,70	0,67	0,49	0,50	0,47	0,44	0,41	0,42	0,43
	19	0,56	0,52	0,55	0,47	0,47	0,50	0,47	0,42	0,53	0,48
	23	0,51	0,58	0,49	0,46	0,50	0,45	0,43	0,43	0,36	0,35

Table 9 present the effectiveness of the ML services from Microsoft for sentences with word counts of 12, 19 and 23 for each noise algorithm for all noise levels. For the no-noise scenario, we have that the Microsoft provider’s f-measure for sentence lengths 12, 19 and 23 are, respectively, 0.72, 0.56 and 0.51. Next, we will perform a noise-to-noise analysis by performing observations.

Keyboard. Considering the **Keyboard** noise, despite observing a lot of variation in the results, the provider does better with length 19, this is more visible at higher noise levels. Also, the provider has worse results with length 23.

OCR. Regarding **OCR** noise, there is little f-measure variation. For this noise, the provider seems to be good for sentences of length 12 in low noise scenarios, while for sentences of length 19 it is better for higher noise levels.

RandomCharReplace. For **RandomCharReplace** noise, at initial noise levels the provider does better with length 12 and 19 and worse with length 23. Disregarding the sudden increase in f-measure for length 12 at noise level 60%, the noise stabilizes for all sentence lengths at 50%.

CharSwap. In CharSwap noise, up to 40% noise, the provider is better in sentences of length 12 and worse in sentences of length 23. After that, the results vary a lot without stabilization.

WordSwap. Regarding **WordSwap** noise, the provider does best at sentence length 12. The other sentence lengths have similar f-measure values. For all of them, the biggest f-measure drop is at higher noise levels.

WordSplit. For **WordSplit** noise, in general, there is little f-measure variation. Within the perceived variations, we see that for this noise, the provider is slightly better for sentence length 12 at low noise levels. In other sentence lengths, the provider performs similarly.

Antonym. Considering **Antonym** noise, the provider is best for sentence length 12, as in all noise levels the f-measure is best at that sentence length. Furthermore, the provider does worse on sentences of length 19 as it has more points where the f-measure is lower.

Synonym. Similarly to **Antonym**, with the **Synonym** noise, the provider is best for sentence length 12, as in all noise levels the f-measure is the best. Similarly, the provider does worse on sentences of length 19.

Spelling. For **Spelling** Noise, the provider is best for sentences of length 12, this is mostly at initial noise levels, up to around 40%. Also, the provider does worse on sentences of length 19, particularly at higher noise levels, starting at 30%.

TfIdWord. Regarding TfIdWord, the provider also does better at sentence length 12, specially at the initial noise levels, but, for this noise, the provider does worse at length 23.

WordEmbeddings. Considering **WordEmbeddings** noise, the provider does best at sentence length 12, where the provider does best at most noise levels. In the other sentence lengths, the provider performs similarly, but the sentence length with the most bad results is 19.

ContextualWordEmbs. For **ContextualWordEmbs** noise, at initial noise levels the provider does best at sentence length 12. For higher noise levels (greater than 50%) it is best for sentences of length 19. On the other hand, the provider does worse at length 23.

Considering our analysis in the previous paragraphs, we can see that the Microsoft provider does better with sentence length of 12. We arrived at this conclusion by looking at the scenarios where the provider was described as performing better considering sentence length. On the other hand, the worst sentence length for Microsoft was 19, although it was also bad in some scenarios with length 23. We can say this because we had a higher occurrence of noise where length 19 was considered the worst.

Summary: The results indicate that Amazon has better effectiveness with sentence lengths of 12 and 19. Meanwhile, Google and Microsoft do better with sentences of length 12. On the other hand, Google's effectiveness is worse with sentences of length 23 and Microsoft with length 19.

6 Threats to Validity

The validity of the results exposed in this work depends on some key pieces. Inconsistencies or weaknesses in these pieces pose threats to the work. Known threats are listed below:

- **The representativeness of the chosen services:** In this work, we reached conclusions about which MLaaS provider to use in certain circumstances. However, the work is limited to currently testing only the Sentiment Analysis service. This limitation can lead to hasty conclusions about which provider is better or worse since the current scenario is limited. In the future, it is expected to include other services to diversify the analysis and increase the resilience of the results.
- **The resilience of the chosen dataset:** The dataset chosen was the *Twitter US Airline Sentiment*, since sentences were obtained from the social network Twitter, there is a risk of dirty sentences or sentences that already have naturally noise, which could affect our control over the amount of existing noise or our benchmark. In addition, according to the dataset page, the sentiment labeling of the sentences was done manually with the help of collaborators, so the results depend on the classifications performed, wrong classifications can influence the results.
- **The relevance of the chosen noises:** The relevance of this work is highly related to the relevance of the chosen noises. Since we want to bring insights into the use of MLaaS in the real world, the noise needs to be relevant to machine learning practitioners. Noises that may occur in real scenarios have a higher value in our analyses.
- **The dependency on the nlpaug library:** In this work, noises from the Python library nlpaug were selected. This brings us questions about the resilience of noise implementations. In addition, some noises that rely on Deep Learning algorithms required pre-trained models that need to be obtained separately. The selection of models in conjunction with the implementation of noise may jeopardize the resilience of the results of this work.

7 Conclusion

We analyze the influence of different types of noise on the effectiveness of Cloud NLP services provided by Google, Microsoft, and Amazon. To do that, we used a dataset containing sentences annotated with sentiments positive, negative, and neutral. Also, we applied 12 types of noise involving syntactic and semantic ones.

The results indicated that Google is more resilient to syntactic noises and Microsoft is more resilient to semantic noises. Regarding sentence lengths, Amazon has better effectiveness with sentence lengths 12 and 19, and Google and Microsoft with length 12. On the other hand, Google's and Microsoft's effectiveness is smaller with sentence lengths 23 and 19 respectively.

In future work, we intend to analyze more types of noise, services, and providers. Also, we intend to explore other evaluation metrics, for example, fairness. Additionally, it is noted that the sentence lengths chosen for RQ 2 are too close, making it difficult to generalize the results. In the future, we intend to run the experiment with longer and more distant sentence lengths. Another important point is a more detailed analysis of the results, especially in RQ 2, since the sentence lengths chosen are close, the results were not very assertive, requiring further analysis using other methods. Finally, particular care must be given to the items in the Threats to Validity section to improve our results.

Bibliography

ARAUJO, J. et al. Decision making in cloud environments: an approach based on multiple-criteria decision analysis and stochastic models. *Journal of Cloud Computing*, v. 7, n. 1, p. 7, Mar 2018. ISSN 2192-113X. Available at: <<https://doi.org/10.1186/s13677-018-0106-7>>.

ARMBRUST, M. et al. A view of cloud computing. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 53, n. 4, p. 50–58, apr 2010. ISSN 0001-0782. Available at: <<https://doi.org/10.1145/1721654.1721672>>.

BAHJA, M. Natural language processing applications in business. In: WU, R. M.; MIRCEA, M. (Ed.). *E-Business*. Rijeka: IntechOpen, 2020. cap. 4. Available at: <<https://doi.org/10.5772/intechopen.92203>>.

BELINKOV, Y.; BISK, Y. *Synthetic and Natural Noise Both Break Neural Machine Translation*. arXiv, 2017. Available at: <<https://arxiv.org/abs/1711.02173>>.

BOUCHER, N. et al. Bad characters: Imperceptible nlp attacks. In: *2022 IEEE Symposium on Security and Privacy (SP)*. [S.l.: s.n.], 2022. p. 1987–2004.

CHOWDHARY, K.; CHOWDHARY, K. Natural language processing. *Fundamentals of artificial intelligence*, Springer, p. 603–649, 2020.

DALIANIS, H. Evaluation metrics and evaluation. In: _____. *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Cham: Springer International Publishing, 2018. p. 45–53. ISBN 978-3-319-78503-5. Available at: <https://doi.org/10.1007/978-3-319-78503-5_6>.

DEVLIN, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. Available at: <<http://arxiv.org/abs/1810.04805>>.

ERICKSON, B. J.; KITAMURA, F. Magician’s corner: 9. performance metrics for machine learning models. *Radiology: Artificial Intelligence*, v. 3, n. 3, p. e200126, 2021. Available at: <<https://doi.org/10.1148/ryai.2021200126>>.

ERYIGIT, G.; CELIKKAYA, G. Use of nlp techniques for an enhanced mobile personal assistant: The case of turkish. *International Journal of Intelligent Systems and Applications in Engineering*, v. 5, n. 3, p. 94–104, Sep. 2017. Available at: <<https://ijisae.org/index.php/IJISAE/article/view/510>>.

FELLBAUM, C. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998. Available at: <<https://mitpress.mit.edu/9780262561167/>>.

GALBUSERA, F.; CASAROLI, G.; BASSANI, T. Artificial intelligence and machine learning in spine research. *JOR SPINE*, v. 2, n. 1, p. e1044, 2019. Available at: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/jsp2.1044>>.

GIACHOS, I. et al. Inquiring natural language processing capabilities on robotic systems through virtual assistants: A systemic approach. *Journal of Computer Science Research*, v. 5, n. 2, p. 28–36, Apr. 2023. Available at: <<https://journals.bilpubgroup.com/index.php/jcsr/article/view/5537>>.

GOODMAN, D.; XIN, H. *Attacking and Defending Machine Learning Applications of Public Cloud*. arXiv, 2020. Available at: <<https://arxiv.org/abs/2008.02076>>.

GRANA, J. Perturbing inputs to prevent model stealing. In: *2020 IEEE Conference on Communications and Network Security (CNS)*. [S.l.: s.n.], 2020. p. 1–9.

GRANDINI, M.; BAGLI, E.; VISANI, G. Metrics for multi-class classification: an overview. *ArXiv*, abs/2008.05756, 2020. Available at: <<https://api.semanticscholar.org/CorpusID:221112671>>.

JANG, J. et al. Predicting personality and psychological distress using natural language processing: A study protocol. *Frontiers in Psychology*, v. 13, 2022. ISSN 1664-1078. Available at: <<https://www.frontiersin.org/articles/10.3389/fpsyg.2022.865541>>.

KALAPANIDAS, E. et al. Machine learning algorithms: a study on noise sensitivity. In: *Proc. 1st Balcan Conference in Informatics*. [S.l.: s.n.], 2003. p. 356–365.

MA, E. *NLP Augmentation*. 2019. <https://github.com/makcedward/nlpaug>.

MORADI, M.; SAMWALD, M. Evaluating the robustness of neural language models to input perturbations. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021. p. 1558–1570. Available at: <<https://aclanthology.org/2021.emnlp-main.117>>.

MÄNTYLÄ, M. V.; GRAZIOTIN, D.; KUUTILA, M. The evolution of sentiment analysis—a review of research topics, venues, and top cited papers. *Computer Science Review*, v. 27, p. 16–32, 2018. ISSN 1574-0137. Available at: <<https://www.sciencedirect.com/science/article/pii/S1574013717300606>>.

NÁPLAVA, J. et al. Understanding model robustness to user-generated noisy texts. In: *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*. Online: Association for Computational Linguistics, 2021. p. 340–350. Available at: <<https://aclanthology.org/2021.wnut-1.38>>.

PAJOLA, L.; CONTI, M. Fall of giants: How popular text-based mlaas fall against a simple evasion attack. In: *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. p. 198–211. ISBN 9781665414913.

PALLAS, F.; STAUFER, D.; KUHLENKAMP, J. Evaluating the accuracy of cloud nlp services using ground-truth experiments. In: *2020 IEEE International Conference on Big Data (Big Data)*. [S.l.: s.n.], 2020. p. 341–350.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Empirical Methods in Natural Language Processing (EMNLP)*. [s.n.], 2014. p. 1532–1543. Available at: <<http://www.aclweb.org/anthology/D14-1162>>.

RIBEIRO, M.; GROLINGER, K.; CAPRETZ, M. A. Mlaas: Machine learning as a service. In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. [S.l.: s.n.], 2015. p. 896–902.

RYCHALSKA, B. et al. Models in the wild: On corruption robustness of neural nlp systems. In: *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part III*. Berlin, Heidelberg: Springer-Verlag, 2019. p. 235–247. ISBN 978-3-030-36717-6. Available at: <https://doi.org/10.1007/978-3-030-36718-3_20>.

SAEED, M. Y. et al. An abstractive summarization technique with variable length keywords as per document diversity. *Computers, Materials & Continua*, v. 66, n. 3, p. 2409–2423, 2021. ISSN 1546-2226. Available at: <<http://www.techscience.com/cmc/v66n3/41092>>.

SCHOUTEN, K.; FRASINCAR, F. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, v. 28, n. 3, p. 813–830, 2016.

SHAROU, K. A.; LI, Z.; SPECIA, L. 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)*. Held Online: INCOMA Ltd., 2021. p. 53–62. Available at: <<https://aclanthology.org/2021.ranlp-1.7>>.

SHEN, Z.; KRIMPEN, H. van; SPRUIT, M. A lightweight api-based approach for building flexible clinical nlp systems. *Journal of Healthcare Engineering*, Hindawi, v. 2019, p. 3435609, Aug 2019. ISSN 2040-2295. Available at: <<https://doi.org/10.1155/2019/3435609>>.

THO, Q. Modern approaches in natural language processing. *VNU Journal of Science: Computer Science and Communication Engineering*, v. 39, n. 1, 2022. ISSN 2588-1086. Available at: <<http://jcsce.vnu.edu.vn/index.php/jcsce/article/view/302>>.

ZHANG, Y. et al. Interpreting the robustness of neural nlp models to textual perturbations. *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.