TEC | Tecnológico de Costa Rica

Maestría en Computación, énfasis en Ciencias de la Computación

Escuela de Ingeniería en Computación

Automatic Spanish Text Complexity Detection in Financial Documents with Low Data Availability

by

Mario Alberto Romero Sandoval

Supervisor:

Saúl Calderón Ramírez

San José, CRC

May 29, 2024

Escuela de Computación
Unidad de Posgrado

TEC | Tecnológico de Costa Rica

ACTA DE APROBACION DE TESIS

Automatic Spanish Text Complexity Detection in Financial Documents with Low Data Availability

Por: Mario Alberto Romero Sandoval

TRIBUNAL EXAMINADOR

SAUL CALDERON RAMIREZ (FIRMA)
Firmado digitalmente por SAUL CALDERON RAMIREZ (FIRMA)
Fecha: 2024.04.22 13:09:29 -06'00'

Dr. Saúl Calderón Ramírez
Profesor Asesor

JOSE MARIO CARRANZA ROJAS (FIRMA)
Digitally signed by JOSE MARIO CARRANZA ROJAS (FIRMA)
Date: 2024.04.19 10:30:30 -06'00'

Dr.-Ing. Jose Mario Carranza Rojas
Profesor Lector

FABIAN ENRIQUE FALLAS MOYA (FIRMA)
Digitally signed by FABIAN ENRIQUE FALLAS MOYA (FIRMA)
Date: 2024.04.16 11:12:44 -06'00'

Dr. Fabián Fallas Moya
Lector Externo

LILIANA SANCHO CHAVARRIA (FIRMA)
PERSONA FISICA, CPF-03-0257-0983.
Fecha declarada: 29/04/2024 11:22:30 AM

Dra.-Ing. Lilliana Sancho Chavarría
Presidente, Tribunal Evaluador Tesis
Programa Maestría en Computación

SINAES POST-GRADO ACREDITADO COSTA RICA

22 de marzo, 2024

Acknowledgements

My special thanks to my parents and grand parents, whom support was key during all my education journey to this day. For their support during all the steps and accomplishments. And to my grand parents whom will not witness this milestone in this world, my special thanks for all the values and lesson learned during my life.

Also, I want to give my gratitude to my supervisor during this process Dr. Saúl Calderón Ramírez. As well as the entire team working in the text simplification initiative: Dr. Martín Solís Salazar, Prof. Nelson Pérez Rojas and Dr. Mónica Arias Monge, for their direct or indirect collaboration that made this possible. And a special thanks to Dr. Horacio Saggion, for his help and his team collaboration in the text simplification initiative that made possible the access to 2 data sources used in this research also.

Abstract

Access to information is a fundamental human right in modern society. Nevertheless we all do not have equal access to information, and one reason for that is that we do not understand everything in the same way. Education level, age, disabilities and the cultural context may impact the way that a text is read and understood by the public.

Being able to discriminate between complex and simple segments of text has many applications from improve the efficiency of simplifications systems, to education application helping to determine if a text is appropriate for a given student level and also supervise whether institutions are communicating properly its decisions with the public.

In this work, we will explore different method and techniques for text classification based on the complexity, concretely Spanish text, as well as methods to solve the lack of data in general for the task of Spanish text complexity discrimination. Specifically we will focus on the leverage of existing language models and transfer learning to achieve and measure the impact of augmented data by using synthetic data generation in the problem of text complexity discrimination.

List of publications

Conferences Publications with Peer Review

1. Romero-Sandoval, M., Calderón-Ramírez, S., Solís, M., Pérez-Rojas, N., Chacón-Rivas, M., Saggion, H. (2021). Towards Text Simplification in Spanish: A Brief Overview of Deep Learning Approaches for Text Simplification. In 2022 IEEE 4th International Conference on BioInspired Processing (BIP). doi: https://doi.org/10.1109/BIP56202.2022.10032482. San Jose, Costa Rica.

2. Romero-Sandoval, M., Calderón-Ramírez, S., Solís, M. (2023). Using GPT-3 as a Text Data Augmentator for a Complex Text Detector. In 2023 IEEE 5th International Conference on BioInspired Processing (BIP). doi: https://doi.org/10.1109/BIP60195.2023.10379347. Alajuela, Costa Rica.

Contents

List of Figures

List of Tables

Acronyms

ANOVA Analisis of Variance. 40, 41, 46

BERT Bidirectional Encoder Representations from Transformers. 13, 18, 19, 23–25, 32

BETO Spanish BERT. 28, 34, 35, 43

BLEU Bilingual Evaluation Understudy. 20

CEFR Common European Framework of Reference for Languages (CEFR). 12

ConvNet Convolutional Neural Networks. 31

Deep-RNN Deep Recurrent Neural Network. 14

DL Deep Learning. 3, 9, 11

DNN Deep Neural Network. 13

FSG Fine State Grammar. 10

GPT Generative Pre-trained Transformers. 18, 28, 32–37, 43, 45–47

LLM Large Languages Models. 3, 4, 18, 19, 23, 24, 37–39, 45, 47

LM Languages Models. 18, 19

LSTM Long Short Term Memory. 14, 16, 23, 31

ML Machine Learning. 3, 7, 11, 23

MT Machine Translation. 7

mT5 Multilingual T5. 28, 37, 43, 47

NCES National Center for Education Statistics. 1

NLP Natural Language Processing. 7, 10, 14, 18, 19, 24, 25, 31, 37

NN Neural Network. 13, 24

PIAAC Program For the International Assessment of Adult Competencies. xi, 1, 2

QKV Query-Key-Value. 15

RF Random Forest. 23, 31, 32, 34

RL Reinforcement Learning. 19

RLHF Reinforcement Learning from Human Feedback. ix, 19, 20

RNN Recurrent Neural Network. 11, 14, 16, 19

RoBERTa A Robustly Optimized BERT Pretraining Approach. 12, 13

SAMSA Simplification Automatic evaluation Measure through Semantic Annotation. 22

SVM Support Vector Machine. 24

TF-IDF Term Frequency – Inverse Document Frequency. 31, 32, 34

ULMFiT Universal Language Model Fine-tuning for Text Classification. 25

1. Introduction

1.1 Background

Access to information is nowadays a subject of more than a possibility, but a human right [8]. Society nowadays is a fast paced environment in which making decisions is something that occurs frequently but also something that would change the life of a being in many ways. Therefore, having the access to current and updated information but more important being able to understand it at full, becomes quite relevant for everyone.

This fact is something to worry about once we take into consideration that the ability to understand information is not something established uniformly across the world population. Some disabilities may impact the capability of someone to have equal access to information, as well and literacy levels across the population are affecting in the way that different people understand the same piece of information as measured in [9] by the National Center for Education Statistics (NCES).

Also, as mentioned in [10] differences between a document or text complexity and literacy skills of the reader is identified as a source of bias and inequality. In this example, the authors concluded that it is necessary around 18 years of education to be able to understand the clinical trials description on the site ClinicalTrials.gov, which could introduce a self selection bias on those trials.

As an example, according to the PIAAC in the survey performed in 2012-2014 [11], most of the population does not even form part of the highest levels of the survey (See figure 1.1).

One measure is what we can find in figure 1.2, where is shown that in the US the average literacy level by state is below 4 for all cases, and more generally around 1 and 2 [1]. This is a proof that even for the information that is accessible it is not equally understood by all the population generating inequality in an already competitive and fast paced society.

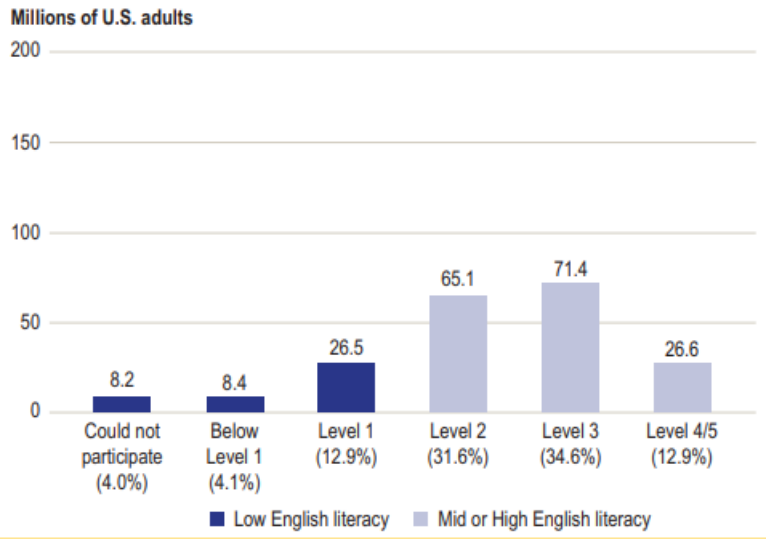

Figure 1.1: Population distribution for the Program For the International Assessment of Adult Competencies (PIAAC) literacy survey 2012-2014

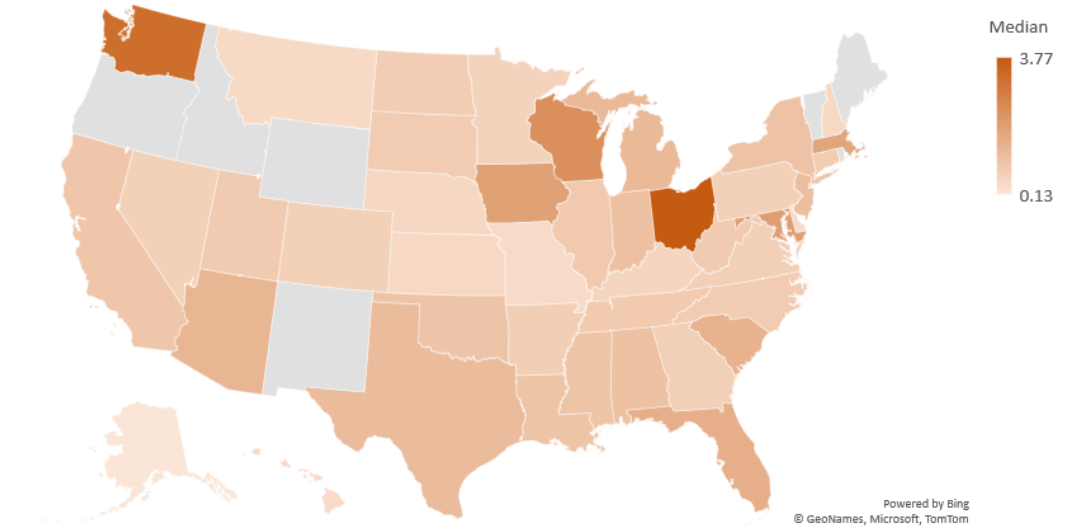

Figure 1.2: Average Literacy level by state according to PIAAC survey 2019 [1]

In general, automated text simplification appears as a solution to this problems by improving fairness, transparency and democratization of information understanding [6]. This methods could provide improvements in areas like education [12], healthcare [13] and also public institutions accountability [14].

1.2 Problem Definition

Automatic text complexity categorization, mostly the current efforts based on Machine Learning (ML) and Deep Learning (DL) methods have two important factors that could affects its real-world performance. The availability of good quality sentence paired datasets and the size of those datasets, which is shown that is scarce and the few available are equality in all languages.

Also, as a more general problem for text simplification. A metric that could determine that for a given text segment if it could be categorized as simple or complex is not present in the current state of the art for this area. This kind of metric or discriminator could accelerate the development of simplification models by helping with the fast generation of big labeled datasets without the intervention of a human, and also as the execution of Large Languages Models (LLM) is computationally expensive, this kind of classifier could be used as a initial step in a simplification pipeline to save on resource usages, both computationally and economically.

In sections 1.2.1 and 1.2.2 there are more details of why this two factors could be affecting text simplification in general and for Spanish as is the focus of the research.

1.2.1 Datasets

For text simplification, most of the research and investment, has been around English language and English speaking population. Therefore, data for other languages rather than English is not abundant. For Spanish there is relevant work in [15], [16] and [17] where some dataset where created by the authors. Nevertheless the size of those datasets is not near what is available in text simplification in English, neither what is used in other text generation task where huge dataset where used, as in [18] where the training dataset was around 45TB worth of data.

1.2.2 Text Simplification Metrics

In particular evaluation of automatic text simplification is an important part as it allow us to know if we have made progress and the current state of our implementation in comparison with the world. The problem is that as stated in [19], it usually require to apply multiple operations (lexical substitution, sentence split, unnecessary information removal, clause organization, among others) but metrics are design to evaluate a single operation in particular or a single category of them. Therefore each metric will leave some of the other possible operation not evaluated and could mislead as it will not provide a complete picture of the results.

Based on that, some studies like [19] stated that while there are multiple metrics to evaluate aspects of a text segment that could indicate its complexity, there is no complete metric that could determine whether or not a text segment is simple or complex.

Currently, the preferred method to evaluate a text simplicity is by asking human evaluators their score for the text in the areas of grammaticality, meaning preservation, and simplicity. This is important as humans tend to prefer the sentences where multiple simplification operation where applied.

1.3 Objectives

Main Objective: Propose one or more techniques to deal with low data availability for Spanish text complexity detection for financial documents.

Specific objectives:

1. Build and/or identify data sources to train a deep learning model for text complexity detection in Spanish.

2. Measure the impact dataset size and the use of generated segments in the Spanish text complexity detection problem.

3. Propose a data augmentation methodology for a text segment classifier using LLM.

4. Evaluate the effects of fine-tuning classification models with augmented data as a way to improve transfer learning results from using only small datasets.

5. Contrast the results of using pre-trained models, such as BETO/BERT as a base for transfer learning.

1.3.1 Objectives changes

As the first initiative presented in the first version was intended to work around the problem of simplification generation, changes in the objectives were applied to match the new objective of dealing with low data availability for Spanish text complexity detection for financial documents.

The reasoning behind this change in target objectives is mainly related to three areas:

- **Computational resources**: Known text generation model which provide great performance are also tied to high computational and time demands for training. Due to time scope for this work to be completed in 16 months and the lack of a secured access to computational resources, this change was also adviced.

- **Problem scope**: Trying to evaluate the problem of simple text generation would require to perform secondary tasks like, validating the quality of the generated text which will require field experts to do a qualitative analysis.

Due to these reasons presented above, and also recommendations from the first evaluation, the scope of this research was decided to be focus on text segment discrimination based on complexity rather than text generation.

2. Literature Study

2.1 Theoretical framework

As mentioned above, equal access to information and the possibilities to understand that information is a good way to provide fairness, transparency and democratization of information. And this, can be obtain by leveraging techniques like automated text simplification.

Generally speaking, automated text simplification has been studied using Machine Translation (MT) models, concretely monolingual translation models. With the intention to allow the model to learn the pattern and rules of both text with different complexity levels, and therefore automated translation could be achieved.

Definition 1. ***Text Simplification****: Process by which, an input text could be worded so that it became easier to read and understand. Using a set of rules, Natural Language Processing (NLP) techniques or ML models, among others instead of human intervention for the text generation task.*

$$f(s) \rightarrow s'$$

So that:

- *s: is an input text segment that contains one or multiple attributes that can mark it as a complex segment*
- *s': is the output text segment that fulfill the following requirements.*
  - *It is easier to understand for a target audience. This is important as the subjective nature of what complex and simple means.*
  - *It preserves the meaning and all the ideas exposed in the original text segment.*

To understand better this process, these are some of the activities or mechanism by which a text can be simplified and the subsequent requirements of each:

Lexical simplification

It refers to the replacement of difficult words with easier to read and understand words, while preserving their meaning. It can include generating more than one word for each one of the replaced words [20]. Within lexical simplification, we can identify the following sub-tasks [20]:

- Complex words identification: Among the first steps for lexical simplification is the identification of words or terms that can be perceived difficult to understand for a specific audience [21]. Different aspects can be used to measure the complexity of a term. Among them distributional behaviour along the text, the morphological structure, different psychological measures, etc. [21].

- Generation of substitution words candidates and selection: After identifying complex words, a set of word candidates can be generated to replace the most complex terms. The replacement word should retain the semantics close enough to the original complex word, and preserve the sentence meaning [22]. The generated candidate words can be then ranked according to the complexity of the candidate terms, and selected accordingly [23].

Syntactical simplification

It consists in the simplification of the sentence structure. Poorly written texts with technical wording can become confusing to different audiences. Readers may struggle to follow the text, and at some point, lose interest in reading the text [24]. Different ways for syntactical simplification can be implemented, depending upon the rules or data built to feed the model. Several types of syntactic complexity causes can be found: long sentences with a number of component clauses, sentences using passive voice or the usage of anaphora [24]. The following are some tasks involved in syntactical simplification of sentences.

- Sentence alignment: In the context of DL architectures for syntactic simplification, extensive training datasets are needed. In the context of text simplification, this ideally means that a set of complex-simple sentences is needed. When using more widely available text simplification datasets, often only simple-complex text pairs are provided [25]. This calls for the need to first automatically or semi-automatically find the complex-simple sentence pairs. Producing a set of complex-simple sentence pairs can be considered more expensive [25]. These aligned sentences can be then used to train a deep learning model.

- Irrelevant information suppression: A simple task to yield sentences with decreased complexity is the elimination of secondary, redundant or less relevant information [26]. For example, the sentence *The workers were outraged, with a passive anger* can be simplified to: *The workers were outraged*.

- Sentence structure modification: A typical transformation to increase sentence simplicity is the modification of its structure. For example, changing a sentence from a passive form to a subject–verb–object structure simplifies its structure and improves its readability [26].

- Sentence splitting: Long sentences are commonly harder to read. Splitting them might increase its simplicity. For instance, a sentence with a number of clauses linked with conjunctions can be splitted into different number of sentences [26].

Discourse simplification

Syntactic modifications of a text to make it easier to read usually lead to changes in the discursive level. For example, simplification often influences the mechanisms of textual cohesion: suppressing pronouns or some secondary clauses might cut or alter text coherence [26]. However, the process of syntactic simplification might also add pronouns or prepositional phrases that make the text harder to understand. In this way, a series of rules have been proposed to address the task of discursive simplification: replace new or repeated entities, substitution generation, substitution selection and substitution ranking [26]. These

rules have been recently proposed [27] and are based on theories that define referring expressions as markers signaling the degree of accessibility in memory of the antecedent [28].

In the following section 2.1.1 there is a review different approaches to lexical and syntactical simplification.

2.1.1 Automatic Text Simplification Methods

Automatic text simplification methods have evolved with the time and the availability of better tools and NLP techniques currently. Two major approaches that can be identified are rule based methods and data-driven method.

Rule based methods

Rule based methods for text simplification are a set of rules that an expert user defined so that they could be applied to the sentence in order to simplified. Concretely we could divide them into two categories:

- **Rule based Syntactical Correction:** as suggested by its name these methods attempt to reduce the sentence complexity by doing syntactic simplification using a determined set of rules. As an example in [29] the authors used two approaches to implement this method. The first one uses a Fine State Grammar (FSG) as a way to separate sentences in chuncks of words, concretely into verbal phrases and noun phrases, to then apply rules to those chunks (reordering, deletion, split) in order to make the sentence simpler without having to worry about the inner structure of the sentences. Also, the authors applied to tagging methodology to later use those to leverage the dependencies between the sentence sections. This method is also applied in the syntactic simplifier of [30].

- **Dictionary replacement:** This method consist mostly in an straight word replacement based on a dictionary of synonyms, that may receive in some cases parameter of the desired simplification level as an aid to choose the synonym from the dictionary as used in [30] lexical simplifier. This method

is also applied in [31] where lexical paraphrases are use to simplify the text by replacing complex phrases and words.

Data-driven methods

In contrast with rule based method, data driven method use the idea of discovering the patterns within the data to leverage those later in order to simplify the text. Whether it is a dictionary generation using paired data, or a full text generation DL model, they follow the same concept of using the data to dictate those rules. Most of the data driven method could be assign in the two following categories:

- **Lexical Substitution:** This method uses a similar dictionary substitution method, with the difference that it is created via a data driven method. In [32] the author created the lexical substitution dataset using the information from simple English Wikipedia paired with the common English Wikipedia.

- **Simplification as monolingual translation:** With the rise of ML and concretely DL has increased the possibility to use machine translation and concretely monolingual (translation within the same language) as used in [33] and [34]. The idea behind this method is to allow the model (Recurrent Neural Network (RNN), Transformer, Diffusion Model) to learn from paired tagged datasets of simple and complex sentences how to transform or translate a complex sentence into its simple version within the same language. This will focus the problem as a sequence to sequence natural language generation.

Even though, there are advances and research in text simplification, text simplification is still a very under-developed application. This is specially due to the nature that most state of the art techniques are data driven and labeled datasets for text simplification are scarce, and usually small [35]. This situation is worst in Spanish as it is not the usually target languages for the few researches and data generation efforts.

2.1.2 Text classification by complexity

In most of the articles reviewed text complexity classification can be studied in two different lines [36]:

- **Document classification:** This task is intended to label a complete text with a level of difficulty or class. Recent studies have emerged, mainly intended to generate a classifier of text to support the learning process of a new language in students.

  As an example [37] worked on a fine-tune GPT-2 and A Robustly Optimized BERT Pretraining Approach (RoBERTa) on data sets labeled with respect to the standard Common European Framework of Reference for Languages (CEFR) (CEFR) proficiency levels for the Portuguese language. Also, [14] worked on a classifier based on a transformer using text and linguistic features to evaluate and improve the effectiveness of the Bank of Russia communication on monetary policy.

- **Segment classification:** This task is focused in the classification of pieces of text (this is what we will be referring as segments during this document). First efforts in this area came from a statistical background from formulas created nearly three decades ago [38]. Examples of those are the readability formulas presented in [38] and the Flesch–Kincaid readability index generated in 1975 [39] both based on segment properties such as: number of words, number of sentences, amount of syllables and average sentence length, among others.

  Statistical based formulas for text complexity have been criticized because are not enough to cover all the factors that characterize the text complexity [40], [41] and have weak statistical bases [42]. Therefore, in the last years have a number of studies focused on the identification of complex text segments, based on machine learning and deep learning models. Those models can improve the identification of complex text segments [43]. Some of them are based on the training of traditional machine learning models such as Randon Forest, XGboost, Support Vector Machines, Long

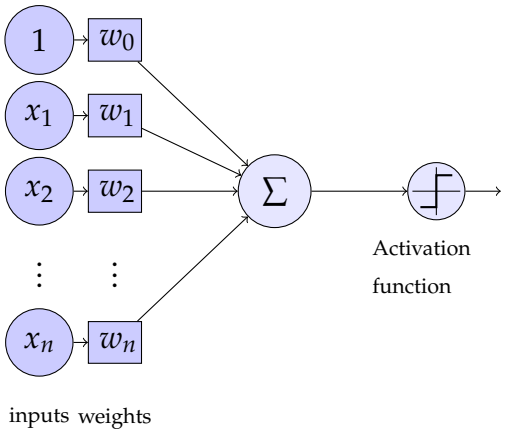

Figure 2.1: Diagram representation of a perceptron

Short Term Memory (LSTM), and others [44], [6], [41]. They use as input features extracted from the sentences, such as linguistic features [45], unigrams, bigrams, and trigrams [6]. Recent models fine-tune pre-trained transformer models like Bidirectional Encoder Representations from Transformers (BERT), RoBERTa, and others, have shown best performance.

Most of literature have focused on languages, such as English, Italian, Russian, and Japanese, and more. Nonetheless, there are no models created for Spanish as far as we know. Therefore, in this work, we use only Spanish text segments.

2.1.3 Neural Networks

Or also called Artificial Neural Network (NN), is a learning model inspired by the biological neurons and their relations. Its most basic design is created in 1958 by Rosenblatt, the perceptron [46]. This design originally created for binary classification evolved into what we currently know as neural networks and later on Deep Neural Network (DNN).

The first design of the perceptron is will operate as a cross product of the inputs and a weight matrix, the result of this operation is later used in an activation function for its final output. The calibration of the weights matrix is usually made using a algorithm called Gradient descent, along with a value known as learning rate.

Definition 2. ***Activation Function****: In NN is usually referred to a function that*

defines the output of a node in the network or the whole network. This function affects the behaviour of the node output.

Definition 3. ***Learning Rate****: Value used in ML algorithms like gradient descent to control the steps in which the weights W are modified in each iteration.*

Definition 4. ***Deep Neural Network (DNN)****: $f(\mathbf{x}, \theta) = y$ with $\mathbf{x} \in X$, y the predicted output and θ the hidden weights for all the neurons.*

2.1.4 Recurrent Neural Network

RNN and Deep Recurrent Neural Network (Deep-RNN) could be descried as models based on the principle that in a sequence the values of the element E_{t-1} affect what are the expected values of the element E_t. This behavior is essentially important in time series predictions and for the nature of language NLP tasks.

On architecture under this category is the Long Short Term Memory (LSTM) networks, first introduced by Hochreiter and Schmidhuber in [47]. This provided great advances in task such as speech recognition, text to voice, as well as other application domains

Definition 5. ***Speech recognition****: Automated task by which a computer algorithm is capable of, given an input in the form of an audio sequence, extract the text of all the dialogues and speech in the input audio.*

2.1.5 Attention

Definition 6. ***Attention (machine learning)****: Technique that tries to mimic cognitive attention. The idea is to increase the importance of some part of the data, while reducing the other ones. Introduced in [48] by Vaswani et al.*

In general, attention is a technique is a technique that came as an improvement to traditional word embedding methods. It calculates ”soft” weights for each word, the difference in this soft weights is that they can change in runtime in contrast to hard weight that are fine-tuned to be frozen for later use.

The main intention of this technique is to take advantage of the hidden layers of the network, in contrast RNN tends to favor more recent information at the

end of the sentence being processed while previous token are expected to be attenuated. Attention allows the calculation equal access to any part of the sentence.

Concretely, attention networks are designed to identify the highest correlations between words in a sentence (previously learnt from corpus). So that an attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, the key and the values are all vectors [48].

Query-Key-Value (QKV)

This concept is an analogous to retrieval systems (ex databases). For example, where you search for file in your file system, it will map your query (text in the search bar) to a set of key (file names) and present you with the best matches (file and it location).

Concretely we can define attention using the following formula.

$$Attention(q, D) = \sum_{i=1}^{m} \alpha(q, k_i) v_i$$

where,

- **q**: is the query begin used.
- **D**: is the database of key-value pairs.
- $\alpha(q, k_i)$: are scalar attention weights.

Describing the process vectorized, given a set of words X, with x_1, x_2, x_3, x_4 words in that set. We can perform the following calculations to demonstrate the attention mechanism.

To begin with, we need to define the matrices K_w, Q_w and V_w which are the weights usually set during training. These weight matrices are used to generate the Key, Query and Value matrices as shown below:

$$K = X \times K_w$$

$$Q = X \times Q_w$$

$$V = X \times V_w$$

Having K, Q and V defined we will calculate:

$$Z = softwax\left(\frac{Q \times K^t}{\sqrt{d^k}}\right) V$$

Where Z matrix is the final output of the self-attention layer. And d_k is the dimension of the matrix of Keys K.

Example: As an example, given a text segment $w =$ "I live in Costa Rica". Our first step is going to be to calculate Q and K which can be described as:

- K vector: How the word or token is described.
- Q vector: What the word of token is looking for in terms of the other token features.

Based on that we are going to calculate the next part of the formula:

$$Q \times K^t$$

which result is a single scalar per token witch for an specific token's "query" the match level with all the other tokens "keys" (referring to vector Q and K respectively). Which can be called, attention vector for a given word which query is being evaluated.

Then we attempt to calculate, vector V for each word which represent the output for the word in given task we are trying to solve. So that we can use our attention vector and the value vector of each word to create a weighted sum (so that the attention value will indicate us the importance for each value vector in the operation) and use it to produce the final output vector Z.

2.1.6 Transformers

Similar to what RNN offer, transformers are design to work with sequences of data. But unlike RNN, transformers use the input all at once instead of doing it element by element. They were introduced alongside the idea of 6 in [48], as a direct replacement for LSTM. From this idea there is already famous implementations such as [18].

Definition 7. ***Encoder - Decoder Architecture****: Section of sequence to sequence architecture design to generate a intermediate result based on a input. This result will be like a synthesis version of the input, that later could be use to generate the output in the decoder. The principle is that the vector will contains valuable features extracted by the encoder so that the decoder will be able to understand and produce an appropriate response.*

Definition 8. ***Encoder Architecture****: Only uses the encoder of a Transformer model. The attention layers can access all the words in the initial sentence. These models are often characterized as having "bi-directional" attention, and are often called auto-encoding models.*

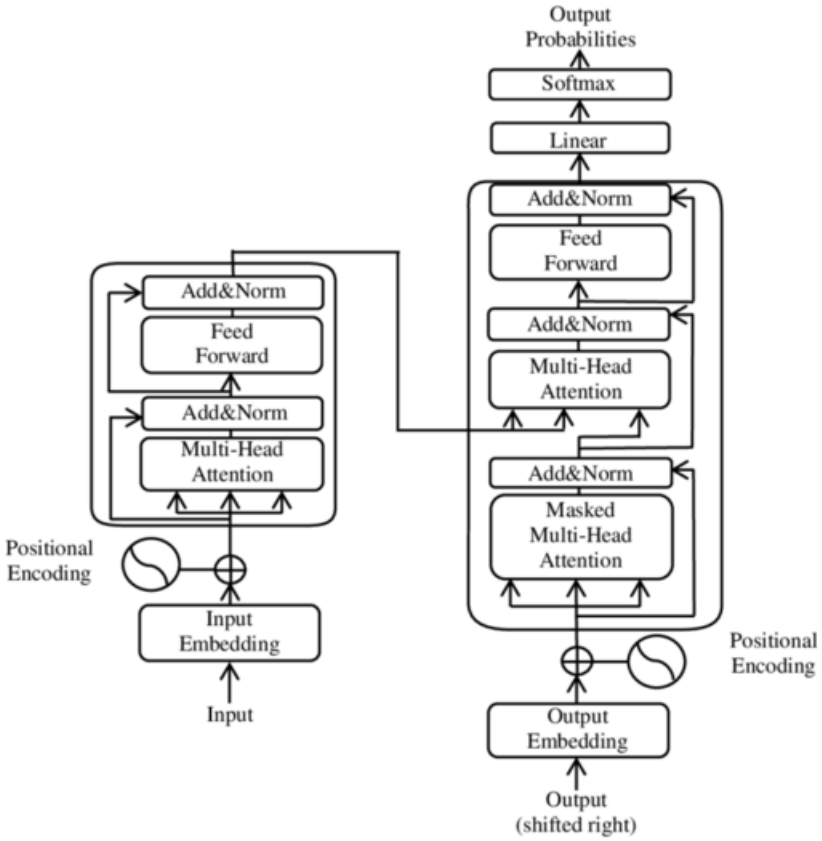

Figure 2.2: Transformer base architecture

The original architecture consist in a Encoder and Decoder 7 design. The encoder consists in a single layer that process the input and generates an encoding that represents the relations of every element of the sequence with the others. But the decoder is a multi-layer design that takes the encoder output with the original input to generate a sequence output.

Definition 9. ***Decoder Architecture****: Uses the decoder of a Transformer model. At each stage, for a given word the attention layers can only access the words positioned before it in the segment. The decoder models usually revolves around predicting the next word in the sentence.*

Large Language Models (LLM) - Languages Models (LM)

Since its first introduction base on the transformer architecture [48] LLM implementation has taken the lead in the advances in many research areas, NLP among them. This also includes generation and classification.

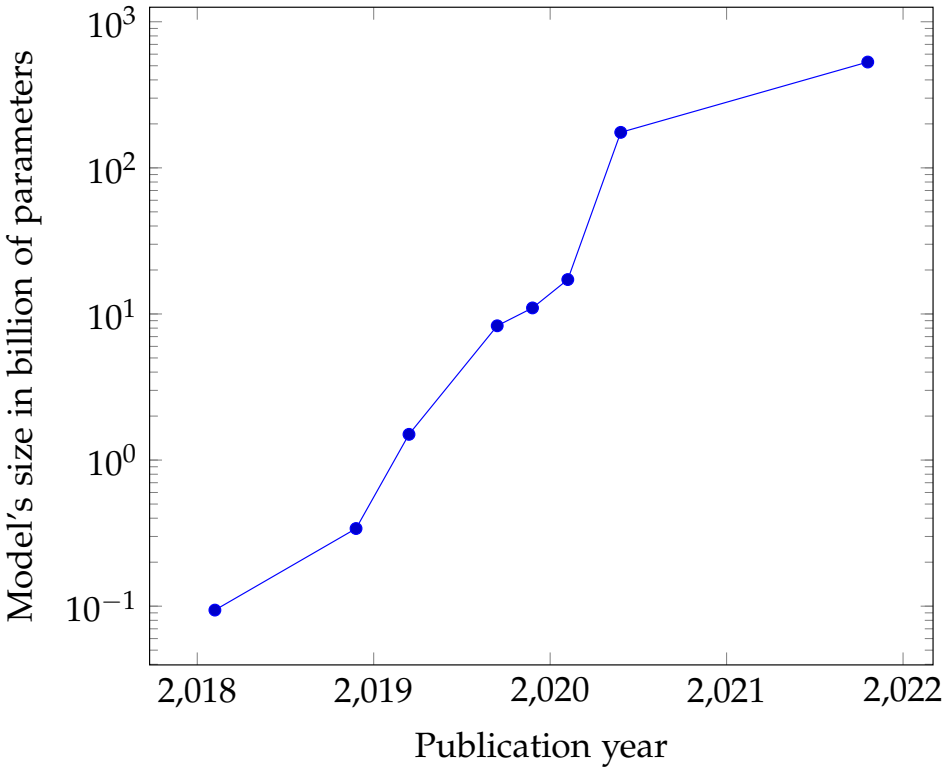

Figure 2.3: Major large languages models (LLM) size tendency from 2018 to 2022 [2] [3] [4]

As shown in figure 2.3, since the introduction of major models like BERT in 2018 [4], there has been a tendency to develop bigger models each time. Most of this driven by the non-stop increase in performance linked to the size of the model, that has driven major breakthrough in many research areas [3].

Currently, most of the latest implementation of these models area not publicly available like Generative Pre-trained Transformers (GPT)-3 [49] and Google's

LLM called Bard, but others as BERT [4] has been use as a baseline for most researches due to its performance and the fact that it is available open source [50].

Concrete we can define the difference between LLM and LM, by first defining what is a LM which is a probabilistic model of a natural language. This model are useful in multiple tasks such as:

- Speech recognition.
- Machine translation.
- Text generation.
- Information retrieval

In contrast, LLM are a more advanced form of a LM. Which are a combination of large dataset and transformer models which due to the attention mechanism provide better information abstraction and extraction from the large dataset. This transformer based LLM have superseded RNN in this task which previously superseded pure statistical models.

2.1.7 Reinforcement Learning from Human Feedback (RLHF)

RLHF is a method used to train or align a model with human preferences or choices. In traditional Reinforcement Learning (RL) the goal is that the model learn a function that matches the behavior of the expected final state. Whereas in RLHF, because in term of human preferences it tends to be difficult to define explicitly a reward function that approximates human preferences, therefore it seeks to train a "reward model" directly from human feedback.

As stated in [51], human feedback is desirable when a task is difficult to specify yet easy to judge. As an example, generating text could be a time demanding and difficult task to assign a group of human experts, but judging AI generated text is easier and still preserve the human input.

In NLP, RLHF has been use because of the difficult nature of defining and measuring a reward model in NLP tasks, as RLHF can drive NLP models, in particular language models providing answers that align with human preferences [51].

RLHF has some limitations with collecting human feedback, learning a reward model, and optimizing the policy [52]. In general:

- **Data Collection:** the scalability and cost of human feedback can be slow and expensive.
- **Reward Model:** The effectiveness of the technique is totally dependant of the quality of the human feedback. The model may become biased or impartial if the feedback provided lacks impartiality, is inconsistent, or incorrect.
- **Optimizing:** A model could learn to exploit the fact that it is rewarded for what is considered positively and not necessarily for what is actually good, which can lead to it learning to persuade and manipulate.

2.2 State of the Art

2.2.1 Text Complexity Metrics

In order to measure text simplification effectiveness and to compare results with other model or methods. The ideal method to evaluate is to present the results to a human evaluator and to assign scores based on the values of simplicity, grammatically correctness and meaning preservation. Nevertheless, some automatic metrics have been used to evaluate results in a easier and faster way. The following are some of the most by some state of the art models.

Bilingual Evaluation Understudy (BLEU)

This metric was originally created for the problem of machine translation. Was created under the principle that: ”the closer a machine translation is to a professional human translation, the better it is” [53].

This score is calculated by evaluation the generated or purposed text segment (usually sentences) against a good quality reference sentences. Evaluating how close the candidate sentence is to the reference. Intelligibility or grammatical correctness are not considered when evaluating that.

This score has high correlation with human judgment. It is expressed by a number between 0 and 1, of how similar the candidate is to the reference sentence. Therefore good quality references sentences are important for its accuracy and correctness.

The score calculation can be defined as:

$$BLEU_w(\hat{S}; S) := BP(\hat{S}; S) \cdot \exp\left(\sum_{n=1}^{\infty} w_n \ln p_n(\hat{S}; S)\right)$$

with,

- $\hat{S}$: Generated output or candidate sentences
- S: Reference sentence
- $BP(\hat{S}; S)$: the Brevity penalty results of evaluating both sentences.
- $p_n(\hat{S}; S)$: modified n-gram precision

Sari

Sari is a metric aimed to evaluate lexical simplicity, by considering how good words are added, deleted. This metric was introduced in [54] and relays on a good set of reference sentences to evaluate the candidate against.

$$SARI = d_1 F_{add} + d_2 F_{keep} + d_3 P_{delete}$$

with

- $d_1 = d_2 = d_3 = \frac{1}{3}$
- F_{add}: F-1 Score for the add operation.
- F_{keep}: F-1 Score for the keep operation.
- P_{delete}: Precision of the delete operation.

Simplification Automatic evaluation Measure through Semantic Annotation (SAMSA)

Introduced in [55], SAMSA is a metric designed to evaluate text simplification beyond paraphrasing. Meaning that, it will allow us to measure the structural simplicity of the sentence rather than lexical simplicity.

It works under the premise that a simple sentence generation will follow these two points.

- Each sentence contains a single event from the input [55].
- The main relation of each of the events and their participants are retained in the output [55].

	References	Structural Simplicity	Paraphrase Simplicity	Meaning Preservation	Method
BLEU	Yes			X	Compare words
Sari	Yes		X	X	Compare generated sencente with reference
SAMSA	No	X			Compare semantic structure
BERTScore$_p$	Yes	N/A	N/A	N/A	Token similarity using contextual embedding

Table 2.1: Comparison of most used state of the art metrics

Flesch-Kincaid Grade Level

Created under a contract with the US navy in 1975 [39]. This metric evaluate the ease of reading an specific sentence. The final result is expressed in a rage from 0 to 100 with a direct relation with the level of education required to successfully read and understand the text with ease, described as follows:

The formula to calculate this score is based on the number of words, sentences and syllabus of the evaluated text. Concretely is defined as:

$$FK_{gl}(S) = 0.39 * \left(\frac{total\_words}{total\_sencentes}\right) + 11.8 \left(\frac{total\_syllabes}{total\_words}\right) - 15.59$$

Score	School Level
100-90	5^{th} grade
90-80	6^{th} grade
80-70	7^{th} grade
70-60	8^{th} & 9^{th} grade
60-50	10^{th} to 12^{th} grade
50-30	College
30-10	College graduate
10-0	Professional

Table 2.2: Flesch-Kincaid Grade Level score in relation to school level

Due to this, this method is heavily affected by the length of the sentences being evaluated. With a bias to be in favor of short segments over long segments, even if there is less lexical complexity in the long one.

2.2.2 Machine Learning based Text classification methods

Classification is one of the classical usages of ML algorithms and there are many application with Text classification. From unsupervised or semi-supervised models, to supervised model trained with (LSTM neural networks and Random Forest (RF)), and also to the current LLM that lead the charts in performance against previous models.

Unsupervised and semi-supervised Machine Learning Approaches

Due to the fact that for some study areas, there is no data available that is fully tagged by experts or the data is not in big dataset, unsupervised learning has been used a way lo leverage the existence of abundant un-tagged data alongside with the small tagged dataset.

On example of this we found in [56], where the authors tested multiple training methods and used a self-pretraining method being purposed in the paper against a BERT classifier as a baseline. In that research, it was found that Self-Pretraining is either the top model or on a par with the top model.

The previous as a good proof that the potential of unsupervised techniques when data is abundant but not fully tagged.

Supervised Machine Learning Approaches

Supervised learning implementation are the most common method used for classification when there is data available as it is the most common usage of the models. Some implementation for sentence level complexity can be found in [5] and [14].

In [5], the authors evaluated for the Russian language the performance of Support Vector Machine (SVM), BERT and Graph NN, among others but we will focus on the ones with high performance to review.

Model	3 classes	11 classes
SVM	75.78%	44.33%
BERT	**81.99%**	**55.89%**
GNN	73.48%	44.04%

Table 2.3: Supervised training results (f1-score) for a complexity detector from [5].

As shown in table 2.3, the authors obtained results that indicates that BERT was outperforming the other supervised approaches in both experiments for 3 complexity classes and 11 complexity classes respectively.

2.2.3 Large Language Models based Classifiers

LLM has been driven the chart in performance for most tasks when deep learning is applied nowadays [3]. For NLP concretely, BERT [4] has been used for most research related to text classification, due to its nature as an encoder based architecture (See definition 8). Concretely, BERT uses a multi-layer bidirectional transformer encoder to create a higher-dimensional output that represents the input and takes in consideration the context of each world, thus helping it to understand the meaning and extract information from the text segment used as input.

Because of that, BERT based classifier has been used in many text complexity detection tasks. Some of the examples will be evaluated below.

BERT based classifiers

One of the most relevant features of BERT is the fact that it's a pre-trained model. It allow BERT to be trained on massive amounts of text data, such as books, articles, and websites, before it's fine-tuned for specific downstream NLP tasks, including text classification. Doing so, BERT can develop a deep understanding of the underlying structure and meaning of language, making it a highly effective tool for NLP tasks.

In [6], we found that the authors evaluated a text complexity classifier for the datasets: Newsela, WikiLarge, Biendata. As shown in table 2.4, from this research the author obtained results around 80% and even as high as 94.43% (the authors mentioned that Biendata contains paired sentences of scientific papers and articles for a wide public consumption, and that different in the sources may be the reson of the abnormal results) using the BERT model.

Model	Newsela	WikiLarge	Biendata
BERT	77.15%	81.45%	94.43%

Table 2.4: Results of BERT model for complexity detection across three different datasets [6]

.

In the previous article, the authors evaluated other models but BERT was highlighted as the top 3 models for the task of complexity classification in those dataset alongside Universal Language Model Fine-tuning for Text Classification (ULMFiT) [57] and XLNeT [58]

3. Methodology

3.1 Proposed Method

We propose the use of a transformer architecture as a baseline to perform text categorization. For this implementation we are planing to base on BETO [59] which is a Spanish implementation of BERT as out initial model that we want to explore the usage of fine tuning for simplicity categorization so that the model will be able to distinguish between simple and complex sentences.

To expand the idea of fine-tuning we will leverage on transfer learning from already trained Spanish language models like BETO [59] and expand it with domain specific and general dataset of text simplification in order to try to leverage the already extracted knowledge in the language.

The idea of using transformers is based on its capacity to better extract the information from text and understand the language model itself due to the nature of the attention model, that models like GPT3 [49] and its implementation (ChatGPT) have shown achievable thanks to its text generation capabilities.

From that point, we want to expand the possibilities to achieve accurate text categorization based on its complexity and effects of the usage of domain specific data to improve this task, compared against general domain data fine tunes models.

Specifically, defining the process of training a model as:

$$M' = train(M, D_{train})$$

And the evaluation process as:

$$S = test(M', D_{test})$$

Where,

- M: the base model being used, for this experiments we will mainly use Spanish BERT (BETO).
- M': trained version of the model M.
- S: Score and variable to use as reference to validate the results, here we will focus on $f1$-score.
- D_{train}: Dataset used for training.
- D_{test}: Dataset used for testing, 20% of the manual dataset. See dataset 1

Our goal will be to create a training dataset D'_{train} so that:

$$D'_{train} = concat(D^i_{train}, D^k_X), 0 < i \leq N \text{ and } 0 \leq k \leq M$$

Where,

- X: Source used to generate the augmented simplifications (GPT-3, Multilingual T5 (mT5), Tuner).
- i: Iteration of the seed dataset size being used. See chapter 5 for reference.
- k: Iteration of the augmented dataset size being used. See chapter 5 for reference
- D^k_X: Augmented dataset used for the iteration of size k and source X

Which will allows to create S' in the form of

$$M'_X = train(M, D'_{train})$$

$$S'_X = test(M', D_{test})$$

So that we can measure if S'_X is comparable with S for different seed and augmented dataset sizes.

3.1.1 Datasets

For these experiments there are two paired sentences dataset that are going to be used for its development. In general the manual dataset or seed dataset was developed in [7] with the target population set as people with visual impediment so that the simplifications are generated accordingly. In terms of the domain, all the complex text segments where selected from financial education documents.

Seed datasets:

1. **Custom financial documents dataset - Manually simplified:** We use an in-house dataset generated from financial education texts [7], where the complex text segments are extracted. These text segments are usually 1 to 2 sentences long. The simplified text versions are generated by 6 human labelers, that use a set of simplification rules based on 21 attributes that indicated if a segment was simple or complex. An example of the three most common attributes can be found in table 3.1. A total of 5314 pairs of complex/simple text segments were generated.

Augmented datasets:

1. **Custom financial documents dataset - GPT-3:** Because of the surge and relevance that tool ChatGPT by OpenAI has achieve in the last weeks we decided to evaluate the possibility to use the tool to generate the simplifications instead of the manual generation made by the expert taggers as a alliterative version of the same source dataset.

   To achieve this task we provide the tool the original complex sentence and the task to provide a simplified version of the segments, this was done without any other input such as custom simplification rules or context in a way that we leave to the model to provide the best simplification it can provide out of the box.

2. **Custom financial documents dataset - mT5:** Same as the previous point. it is based on the manually generated dataset of financial documents. The

Attribute	**Original Text Segment**	**Simplification (Manual)**
Unnecessary Words	No te olvides de las facturas que debes pagar cada pocos meses, por ejemplo, la del seguro del auto	No olvides pagar las facturas de cada pocos meses.
Sentence Length	Otra razón para que este seguro se incremente y se vuelva una necesidad, es el alto número de accidentes vehiculares que muchas veces no sólo afectan al conductor y a su familia, sino a otras personas que son víctimas de ellos.	Otra razón para que este seguro incremente y se vuelva una necesidad, es el alto número de accidentes vehiculares. Estos solo afectan al conductor y a su familia, sino a otras personas que son víctimas de ellos.
Complex Phrases	Tienes que pedir prestado para demostrar que puedes hacerlo de forma responsable.	Pides prestado para demostrar que lo haces responsablemente.

Table 3.1: Examples of complex segments with the most common attributes, and its simple version [7]

.

mT5 automatic simplification model was tested for English text simplification in [60]. This model is a multilingual pre-trained transformer known as T5 [61].

3. **Custom financial documents dataset - Tuner:** This is another source of augmented simplification for the same set of complex sentences. Tuner simplification is a rule-based lexical simplification system with a particular focus on languages like Spanish, Portuguese, Catalan, and Galician [62].

Target datasets:

1. **Custom financial documents dataset - Manual:** It is a subset of 20% of the manual seed dataset (See 1). This subset will only be used for testing and for each experiment, the segments selected are not part of the other datasets.

3.1.2 Models:

During the development of text simplification as a field of study, there has been cases where classification has been used as part of the data prep process. We intent to use a combination of novel model and techniques as well as baseline method already tested to check behaviour and effects of the dataset on those, in order to solve the research questions.

1. **Convolutional Neural Networks (ConvNet):** As stated by [63] [64] a convolutional neural network can be used to generate a representation of the sentence that can be latter processed with clustering to try to extract information from the representation and properly create groups with the vectors that matches the desired characteristics.

2. **RF and Term Frequency – Inverse Document Frequency (TF-IDF):** As a implementation of the RF with TF-IDF can establish a baseline implementation that we can evaluate and compare most current models against. It has been used for text classification

3. **LSTM:** Even that LSTM model are being replaced by attention based model in NLP task. Most of the background research that is not done using a variation of transformers is done using a LSTM network. So its consideration in these experiments is important to establish a baseline of comparison by replicating previous efforts in sentence classification.

4. **BERT:** First exposed by [48] the attention based model and transformers are the state of the art method that is being explored in different areas of NLP, from text generation [49] to classification. These models have achieve great advances in many of NLP areas because of its ability to extract information

and the relation between the words in the whole sentence, capability that can be useful for the complexity classification task. For this model, we will base on BETO [59] which is a Spanish implementation of BERT

Based on the results shown in [6] we will focus on the usage of BERT and RF-TF-IDF, being RF as a baseline for the performance comparison.

3.2 Research Questions

1. Is data augmentation a valid option to solve the lack of Spanish pair complex-simple sentences datasets? How does it compare to building a model from scratch? (See objectives: 2, 4)

2. How does synthetic data (text translation, rule based simplification) affects the current state of the art models for text complexity categorization? (See objectives: 3)

3. Is complex-simple sentence classification something feasible with a small dataset, using our manually simplified dataset, and augmented as a way to solve the limits of the dataset? (See objectives: 1, 5)

3.3 Hypothesis

Based on the research questions, in this document we want to explore the possibilities that:

1. Transfer learning, on pre-train models, using augmented data will provide improvements in model performance ($f1$-score), against models trained with small but manually generated data. (See research questions: 2, 3)

2. GPT-3 text generation capabilities could be used as a source of augmented text to deal with low data availability of complex-simple tagged text segments pairs, providing similar results when using it to fine-tune models. (See research questions: 1, 3)

4. Impact of data quality and dataset size in Spanish text complexity classification

4.1 Introduction

This experiment is design to meet the expectation presented in objective 1, in section 1.3.

> *Build and/or identify data sources to train a deep learning model for text complexity detection in Spanish.*

As well as objective 2, in section 1.3.

> *Measure the impact dataset size and the use of generated segments in the Spanish text complexity detection problem*

And objective 5

> *Contrast the results of using pre-trained models, such as BETO/BERT as a base for transfer learning.*

The hypothesis to validate in this scenario is:

> **Hypothesis 1**: *GPT-3 text generation capabilities could be used as a source of augmented text to deal with low data availability of complex-simple tagged text segments pairs, providing similar results when using it to fine-tune models.*

The previous hypothesis 4.1 is the alternative hypothesis, where GPT-3 could bring results that are statistically similar between using manually generated dataset and automated datasets. The null hypothesis brings the case where there is no similarity between the set of results.

4.2 Experimental Design

To validate the previous hypothesis, we focus on measuring the relative performance of two different classifiers models (BETO and RF, the last one using TF-IDF), this will be done using a performance metric as our target variable ($f1$-score).

To execute this evaluations we aim to train the 2 models with the 4 different simplification sources for the financial segment dataset: manual simplification, GPT-3 generated, mT5 generated and Tuner (rule base simplification).

For this, we will perform a split of 90%-10% for training and testing data respectively, which will bring the total number of segments for each subset to 531 text segment pairs for testing and 4781 text segment pairs for the training subset. Besides that also a K-folds technique is used to perform cross validation, for this we a using a value of $k = 10$ so that each combination of parameter of the experiment will be repeated 10 times for this process.

Dataset	Number of Models	Datasets Sizes	K-fold size	Total Executions
Manual	2	20	10	400
GPT-3	2	20	10	400
mT5	2	20	10	400
Tuner	2	20	10	400

Table 4.1: Distribution of the tests for the experiment detailed in this chapter (4) explaining the total number of execution for each dataset.

Also, to measure the impact of the dataset size into the model output we will also run the evaluation with subsets of the full data. Concretely, we will start using 5% (239 segments pairs) of the training data and perform increments of 5% until we reach the 100% of the training dataset. Concretely, we can see the experiment cases distribution in table 4.1.

4.3 Results

Based on the results provided by the BETO model, in figure 4.1, where we compare the f1-score of the results after training the model with the 4 different datasets over the 10 partitions. We found that following our intuition, the two best dataset that can be used to train the model are the manually generated and tagged and the one generated by GPT-3.

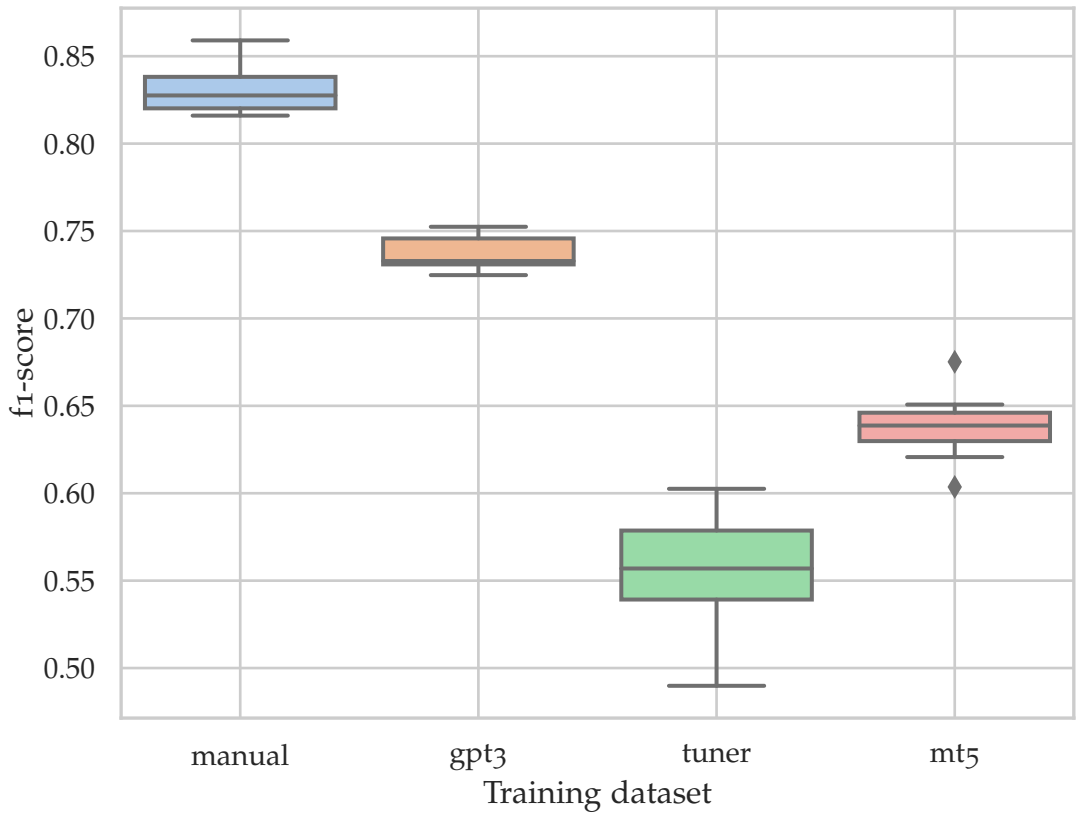

Figure 4.1: Relative performance (f1-score) of the 10 fold cross validation for the 4 different sentence simplification sources (using BETO)

These results provided a clear statistical difference (0.99 p-value) between all the models, which provides evidence of the importance of good quality simplification for the datasets.

In term of size effects, for BETO based model we executed the experiment that produced the results of figure 4.2. There we can explore the effects of training dataset sizes for the 4 different simplification sources. Here we can see that

besides the clear saturation of the model on the non-manual datasets after 2000 segments, there is still capacity to keep increasing the model performance (f1-score) on the manual dataset until near 4000 segments. These behaviour indicate that besides the clear difference in f1-score of the 4 datasets there is also structural or semantic difference in the generated simplified segments as the manuals do not reach a saturation point as fast as the other 2 closest datasets (GPT-3 and mT5).

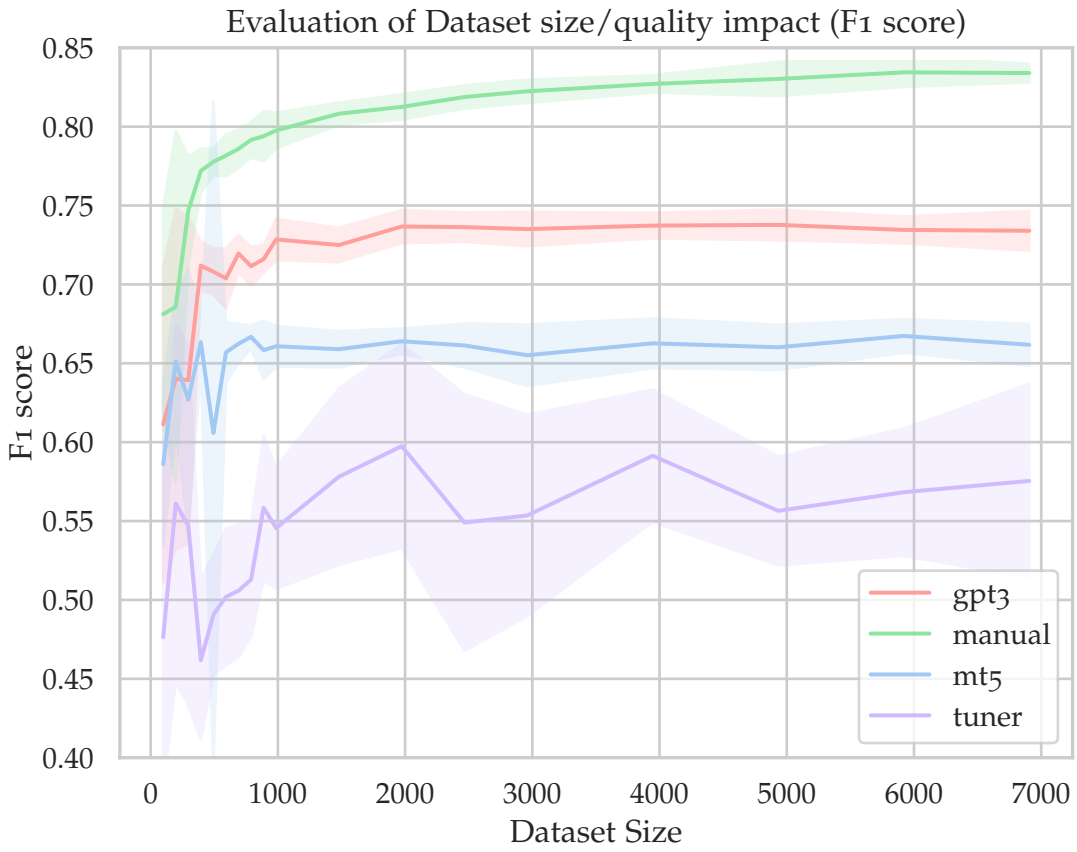

Figure 4.2: Effects of dataset size in model performance (f1-score) for the 4 different sentence simplification sources (using BETO)

4.4 Conclusion

From the previous results we can conclude two main points. First, we can conclude that there is a clear advantage to use deep models (concretely transformers), since the two methods using this technology (mT5, GPT-3) obtained better

results that rule base generation. Specially with GPT-3 we can produce results from our predictor that are close to the ones using manual data as our training source.

From that point, related to research question 1, as shown in figure 4.2, using LLM text generators as a way to get synthetic data allow us to produce results that are close to what we can achieve using only data generated by experts. This, even that does not show that it is comparable to expert data, points out two things:

- Synthetic data quality generation has been increasing in performance from manual methods (Tuner) to ML methods (mT5, GPT-3), which shown a clear tendency to improve so that we can expect to get closer as more complex and bigger models are develop. As this is also the tendency that most LLM based solution are experiencing in many fields [3].

- Even that the results are not statistically comparable in all dataset size, for some dataset (specially on the lower size end), GPT-3 generated simplifications can be used as valid precursor for research and proof of concept development. Which allow faster startups and iterations in research initiatives and projects in general.

Similar to those results, in different NLP domains, we have found that LLM have been used for data generation providing closer results to what using only human generated data could achieve [65]. Where authors used GPT-2 as a data generator and obtain comparable results as using human generated data, in the field of fine-grained claim detection in financial documents. Besides that, in [65] authors also encounter a similar behaviour that we found where the model relative performance, which increase fast with the first samples and then reduce its rate really fast, like a saturation (see figure 4.2).

Also [66] did a similar approach for hate speech detection using mT5 as a data generator for a single class classifier, with the generated data presenting the best performance and similar to the original dataset. Those cases besides having a different domain, share similar properties as our target experiment and

we also observed similar results highlighting the advances in LLM for synthetic data generation.

Nevertheless, for this experimental setup, we are going to reject our hypothesis and accept the null hypothesis as in term of the predictor results ($f1$-score) we can establish a clear statistical difference in the score (p-value > 0.99).

5. Data augmentation impact in Spanish text complexity detection

5.1 Introduction

Here, we aim to explore the objective presented in section 1.3, concretely objective 3:

> *Propose a data augmentation methodology for a text segment classifier using LLM.*

Also, objective 4:

> *Evaluate the effects of fine-tuning classification models with augmented data as a way to improve transfer learning results from using only small datasets.*

Based on that, we are validating the hypothesis:

> **Hypothesis 2**: *Transfer learning, on pre-train models, using augmented data will provide improvements in model performance (f1-score), against models trained with small but manually generated data.*

This hypothesis 5.1 is the alternative hypothesis, where there is statistical difference when using domain specific data for the predictor. The null hypothesis stands for the case where there is no measurable benefit from using domain specific data in transfer learning.

5.2 Experimental Design

To validate the previous hypothesis, we are going to create a process that measures the performance of the model (F1-score), and measure the effects of that performance metric by increasing the augmented dataset size and validate it

with multiple sizes for the seed dataset used in the training. See section 3.1 for reference.

For this, we will use the following configurations:

	Seed Dataset	Augmented Dataset	Total Executions
Configurations	10	16	160

Table 5.1: Distribution of seed and augmented dataset used in the experiment detailed in this chapter (5) explaining the total number of executions.

Concretely, we will be using the following values for the dataset sizes in each experimental configuration:

- Seed dataset size configurations: 97, 196, 294, 393, 492, 985, 1478, 1972, 2465, 2958.
- Augmented dataset size configurations: 0, 48, 146, 245, 344, 492, 738, 985, 1478, 1972, 2465, 2958, 3452, 3945, 4438, 4932

We aim to evaluate the effects of dataset size combination (seed and augmented) in the model performance, to check if there is a correlation of those two values. For this, we execute the following steps for the data augmentation effect experiment:

First, we will begin by executing a combination between seed dataset size and augmented dataset size in a iterative way so that we can measure the changes respective to the previous step pipeline. Then, considering all the 450 experiment executions (this is because is combination needs to be executed 3 times for statistical validity), each will create a data point that will be part of the final results.

Finally, an Analisis of Variance (ANOVA) execution will be perform between specific augmented dataset sizes, for the same seed size, so that we can measure if there is statistical difference between each data point and the tendency over multiple augmented dataset sizes.

5.3 Results

Our main goal for this experiment is to validate if there is a significant difference between the results using a given seed dataset size and after adding some artificially generated simplifications from the augmented dataset.

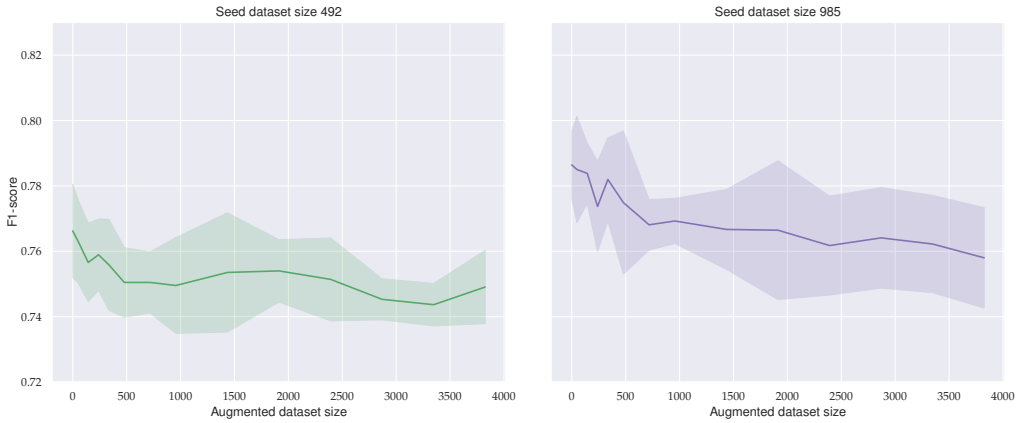

Figure 5.1: Results of executing the data augmentation with 492 and 985 seed dataset sizes

Initially when evaluating with a seed dataset size of 492 and 985 (See figure 5.1), we observe that there is a decrement in model performance with the increase of augmented data. These could be a result of the performance saturation from the model when using seed dataset with the tested size (See table III).

Given such evidence, we focus on the effect of data augmentation when using smaller seed dataset sizes. In such settings, as shown in Figure 5.2 we can observe that the model increases its performance more noticeably when adding augmented data. Specifically with seed dataset sizes of 97 and 196 text segments we can observe that there is a tendency to a F1-score increase with more augmented data. A saturation point around 1500 generated sentences is reached, where the model does not improve its F1-score any further, by adding more augmented data.

When performing the ANOVA test with a $p = 0.05$ we found that for seed dataset size from 196 text segments there is not a significant statistical F1-score increase when using any number of augmented text segments. In the case of a seed dataset size of 97, after adding 478 augmented text segments, we found

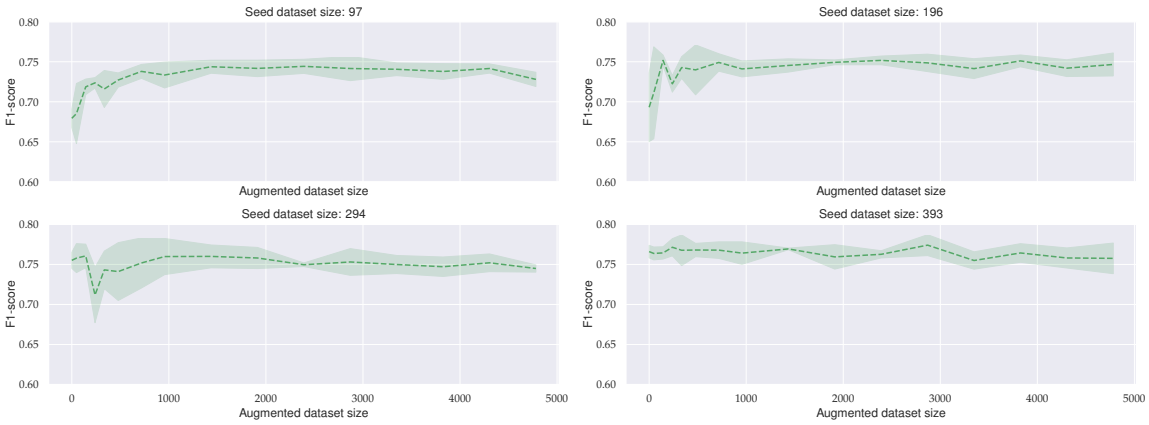

Figure 5.2: Results of executing the data augmentation with the 4 smallest seed dataset sizes

a statistical difference with $p = 0.05$, and this behavior persists for all the augmented dataset sizes after 478. This using the baseline of not performing any augmentation at all.

Finally, we can affirm that this method of data augmentation is only useful for small datasets, around ≈100 text segments, the is essentially caused because as the performance increase of using generated data gets less effective the larger the manual simplified seed dataset is used as part of the training. See figure 5.1 as example of seed dataset sizes are big enough that adding augmented data caused a decrement in model performance ($f1$-score).

5.4 Conclusion

After evaluating the effects of data augmentation and datasets quality in complexity detection, we can recognize that there is a measurable benefit on using augmented data (See section 5.3), but only when we are applying to small datasets.

An example of experiments in this area is [6] where authors develop a classification experiment was executed using 3 dataset with no augmented data. In contrast that experiment used datasets with larger amount of sentences, but achieved similar results for the datasets (excluding Biendata as a outlier, as pointed out by the authors) that we achieve using only manual data. When

evaluating the use of augmented data [67] performed an experiment with multiple dataset that showed small gains of using augmented data, but a decrement from using all the data for fine tuning without any augmentation (similar to what we obtain when using large seed dataset sizes).

From that point, we can provide an answer, specific to complexity detection, to research question 3 as for a given dataset size using augmented data produce us results that, as shown in figure 5.2 and figure 5.1, indicated that even if results are not comparable for all sizes and big datasets where expert data perform better. For smaller dataset, which are the focus of the research question, we find out that the we can achieve small but measurable improvements while measuring our target variable.

That being said, we also measured that the quality of the automatic simplifications greatly influences the performance gain as GPT-3 augmented data showed a much higher accuracy gain when used, compared to less powerful approaches using mT5 and Tuner. A BETO model is not much benefited with the increase of training data, after certain point, even if it is manually labeled data. This suggests that other architectures, loss functions or other modifications can be explored to improve performance, as well as the distribution of data in the sample so that it may be caused by an underrepresented group or attribute from the original dataset.

6. General Conclusions

In this chapter, we summarize the achievements and conclusions inspired by our results. Also a section of the limitations of the research and future work will be covered.

6.1 Main Findings

Currently, the data generation capabilities of LLM are being explored in multiple research projects and areas, an example of those based on GPT is summarized in [68]. As per shown in [68], establishing that LLM generated data is as good as human generated data, could impact multiple fields and areas of the society by accelerating the researches in those areas.

Nevertheless, the usage of LLM generated data for research could face some limitations due to its characteristics, such as issues with ethics [69], biases [70] and hallucinations [71] among others. To solve those issues still research is needed so that it can be controlled or detected so that generated data can be more trustworthy without any external process. Even though it has limitations it impacts in areas that require data augmentation due to imbalance or overall availability, survey analysis, signal generation among other could speed up researched in areas where data availability is an issue due to natural imbalance of the data, cost of collecting the data or data availability at all.

As a summary of the achievements and findings based on the evaluation of the hypothesis in the experiments presented previously we can list the following aspects.

1. In terms of the capabilities of LLM, GPT-3 concretely, to be used to generate synthetic data that can be use to train smaller models like classifiers, as we can see in figure 4.1 there is a clear advantage in using LLMs as a generator when compared to other sources or simplifications. Nevertheless, as we can also see in figure 4.2 the results when using synthetic data are still not

comparable with manual simplifications. This concretely for our dataset and our set of attributes being used as guidelines for simplification.

Also, we could measure a model saturation at ≈2000 segments in training, this saturation level was different for each source (See figure 4.2) so that we can point out that this could be due to some attributes begin under represented and the fact that for the synthetic data sources this under representation of attribute could be more notorious.

2. Related to the results of using GPT-3 generated data as augmented data to improve model performance when only scarce dataset are present for simplicity classification. As we can see in figure 5.1 and figure 5.2 the impact of synthetic data in the performance can be only noticeable when dealing with really small seed dataset. If we decide to increase the seed dataset size we encounter two situations.

   (a) Initially the impact will be no measurable as there will be no statistically difference from the baseline to the model trained with augmented data also. See figure 5.2, where we can see that for the last 2 sizes there is no big change in the tendency of the results, which we also proof using ANOVA with a p-value = 0.05.

   (b) Then, we start measuring negative impact as the model will decreases in f1-score the more synthetic data we introduce. And this effect increase as we also increase the proportion of synthetic data being used. The previous was also measurable using ANOVA with a p-value = 0.05.

In terms of applications of a complexity detector with augmented data, this will indicate that with our dataset or one created in a similar way (using the same attributes and distribution of segments), we can only expect benefits when working with really small dataset. Nevertheless, by itself augmented data could be near the manual dataset $f1$-score value enough so that I could be use as an initial dataset to generate other studies, products and researches.

6.2 Future work and limitations

Bases on the findings presented in section 6.1, we reach the following set of future research paths.

1. As described in the seed dataset definition (See dataset 1), our definition of what is simple or complex rely on the presence of the attributes used in the guidelines to generate the simplifications [7]. One path that we do not cover in this research was to measure the difference in the generations made by humans and by GPT-3.

   Since if the synthetic ones are more similar to the complex version this could affect the model performance as it would be harder for the model to differentiate between the two groups. It will introduce noise by creating more overlap into the two target classification groups (simple - complex).

   We think that this is a possibility as GPT-3 probably will not apply all the simplification rules being used in the manual dataset and that will create underrepresented samples.

2. Other approaches to generate artificial data could also be explored that were not covered in this research like accumulative generation, as a way to explore the effects of mixing the generated dataset into one.

   Also in this paper we explore the idea of generating the simple version of the segments with the data augmentation techniques (GPT-3-3, mT5, Tuner), but there is also the possibility to use LLM to generate the complex version from a simple segment and compare the effects of those generation against the ones used in this article (generated from complex to simple segments).

3. Finally, in last year new method of fine-tuning LLM has been developed which provide different improvements, such as [72] which provide a faster way of training the model due to limiting the number of trainable parameters, but also has proven to improve the performance of given models in downstream task by avoiding that the fine-tuning process disrupts with

the information already learnt from the large corpus used in the original training.

References

[1] S. Mamedova and E. Pawlowski, "Adult Literacy in the United States," 2019.

[2] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018.

[3] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2023.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[5] V. V. Ivanov, "Sentence-level complexity in russian: An evaluation of bert and graph neural networks," Frontiers in Artificial Intelligence, vol. 5, 2022.

[6] C. Garbacea, M. Guo, S. Carton, and Q. Mei, "Explainable prediction of text complexity: The missing preliminaries for text simplification," in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (C. Zong, F. Xia, W. Li, and R. Navigli, eds.), (Online), pp. 1086–1097, Association for Computational Linguistics, Aug. 2021.

[7] N. Perez-Rojas, S. Calderon-Ramirez, M. Solis-Salazar, M. Romero-Sandoval, M. Arias-Monge, and H. Saggion, "A novel dataset for financial education text simplification in spanish," 2023.

[8] Organization Of American States, "The right of access to information," 2009.

[9] NCES, "Piaac - what piaac measures." https://nces.ed.gov/surveys/piaac/measure.asp. (Accessed on 11/08/2022).

[10] D. T. Wu, D. A. Hanauer, Q. Mei, P. M. Clark, L. C. An, J. Proulx, Q. T. Zeng, V. V. Vydiswaran, K. Collins-Thompson, and K. Zheng, "Assessing the readability of ClinicalTrials.gov," Journal of the American Medical Informatics Association, vol. 23, pp. 269–275, 08 2015.

[11] S. Mamedova and E. Pawlowski, "Adult Literacy in the United States," 2014.

[12] E. Mayfield, M. Madaio, S. Prabhumoye, D. Gerritsen, B. McLaughlin, E. Dixon-Román, and A. W. Black, "Equity beyond bias in language technologies for education," in Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications (H. Yannakoudakis, E. Kochmar, C. Leacock, N. Madnani, I. Pilán, and T. Zesch, eds.), (Florence, Italy), pp. 444–460, Association for Computational Linguistics, Aug. 2019.

[13] E. Abrahamsson, T. Forni, M. Skeppstedt, and M. Kvist, "Medical text simplification using synonym replacement: Adapting assessment of word difficulty to a compounding language," in Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR) (S. Williams, A. Siddharthan, and A. Nenkova, eds.), (Gothenburg, Sweden), pp. 57–65, Association for Computational Linguistics, Apr. 2014.

[14] A. Evstigneeva and M. Sidorovskiy, "Assessment of clarity of bank of russia monetary policy communication by neural network approach," Russian Journal of Money and Finance, vol. 80, no. 3, pp. 3–33, 2021.

[15] L. Martin, A. Fan, É. de la Clergerie, A. Bordes, and B. Sagot, "Multilingual unsupervised sentence simplification," arXiv preprint arXiv:2005.00352, 2020.

[16] A. Palmero Aprosio, S. Tonelli, M. Turchi, M. Negri, and A. Di Gangi Mattia, "Neural text simplification in low-resource conditions using weak supervision," in Workshop on Methods for Optimizing and Evaluating Neural

Language Generation (NeuralGen), pp. 37–44, Association for Computational Linguistics (ACL), 2019.

[17] H. Saggion, "Evaluation of automatic text simplification: Where are we now, where should we go from here," 2022.

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in Advances in Neural Information Processing Systems (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020.

[19] F. Alva-Manchego, C. Scarton, and L. Specia, "The (un)suitability of automatic evaluation metrics for text simplification," Computational Linguistics, vol. 47, pp. 861–889, Dec. 2021.

[20] D. Ferrés and H. Saggion, "ALEXSIS: A dataset for lexical simplification in Spanish," in Proceedings of the Thirteenth Language Resources and Evaluation Conference, (Marseille, France), pp. 3582–3594, European Language Resources Association, June 2022.

[21] D. De Hertog and A. Tack, "Deep learning architecture for complex word identification," in Thirteenth Workshop of Innovative Use of NLP for Building Educational Applications, pp. 328–334, Association for Computational Linguistics (ACL); New Orleans, Louisiana, 2018.

[22] Y. Fu, Y. Feng, and J. P. Cunningham, "Paraphrase generation with latent bag of words," Advances in Neural Information Processing Systems, vol. 32, 2019.

[23] O. Melamud, O. Levy, and I. Dagan, "A simple word embedding model for lexical substitution," in Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pp. 1–7, 2015.

[24] M. Shardlow, "A survey of automated text simplification," International Journal of Advanced Computer Science and Applications, vol. 4, no. 1, pp. 58–70, 2014.

[25] C. Jiang, M. Maddela, W. Lan, Y. Zhong, and W. Xu, "Neural crf model for sentence alignment in text simplification," arXiv preprint arXiv:2005.02324, 2020.

[26] A. Todirascu, R. Wilkens, E. Rolin, T. François, D. Bernhard, and N. Gala, "Hector: A hybrid text simplification tool for raw texts in french," in 12th International Conference on Language Resources and Evaluation (LREC), 2022.

[27] A. Todirascu and R. Wilkens, "Simplifying coreference chains for dyslexic children," in The 12th Language Resources and Evaluation Conference, pp. 1142–1151, The European Language Resources Association (ELRA), 2020.

[28] M. Ariel, Accessing noun-phrase antecedents (rle linguistics b: Grammar). Routledge, 2014.

[29] R. Chandrasekar, C. Doran, and B. Srinivas, "Motivations and methods for text simplification," in COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics, 1996.

[30] J. Carroll, G. Minnen, D. Pearce, Y. Canning, S. Devlin, and J. Tait, "Simplifying text for language-impaired readers," in Ninth Conference of the European Chapter of the Association for Computational Linguistics, (Bergen, Norway), pp. 269–270, Association for Computational Linguistics, June 1999.

[31] K. Inui, A. Fujita, T. Takahashi, R. Iida, and T. Iwakura, "Text simplification for reading assistance: A project note," in Proceedings of the Second International Workshop on Paraphrasing, (Sapporo, Japan), pp. 9–16, Association for Computational Linguistics, July 2003.

[32] M. Yatskar, B. Pang, C. Danescu-Niculescu-Mizil, and L. Lee, "For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia," in Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, (Los Angeles, California), pp. 365–368, Association for Computational Linguistics, June 2010.

[33] Z. Zhu, D. Bernhard, and I. Gurevych, "A Monolingual Tree-based Translation Model for Sentence Simplification," 2010.

[34] W. Coster and D. Kauchak, "Simple English Wikipedia: A New Text Simplification Task," 2011.

[35] F. Alva-Manchego, J. Bingel, G. Paetzold, C. Scarton, and L. Specia, "Learning how to simplify from explicit labeling of complex-simplified text pairs," in Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (G. Kondrak and T. Watanabe, eds.), (Taipei, Taiwan), pp. 295–305, Asian Federation of Natural Language Processing, Nov. 2017.

[36] M. Romero-Sandoval, S. Calderón-Ramírez, and M. Solís, "Using gpt-3 as a text data augmentator for a complex text detector," in 2023 IEEE 5th International Conference on BioInspired Processing (BIP), pp. 1–6, 2023.

[37] R. Santos, J. Rodrigues, A. Branco, and R. Vaz, "Neural text categorization with transformers for learning portuguese as a second language," in Progress in Artificial Intelligence: 20th EPIA Conference on Artificial Intelligence, EPIA 2021, Virtual Event, September 7–9, 2021, Proceedings 20, pp. 715–726, Springer, 2021.

[38] J. R. Bormuth, "Development of readability analyses. final report, project no. 7-0052," Contract, vol. 1, 1969.

[39] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," 1975.

[40] A. Cuzzocrea, G. L. Bosco, G. Pilato, and D. Schicchi, "Multi-class text complexity evaluation via deep neural networks," in Intelligent Data Engineering and Automated Learning–IDEAL 2019: 20th International Conference, Manchester, UK, November 14–16, 2019, Proceedings, Part II 20, pp. 313–322, Springer, 2019.

[41] G. L. Bosco, G. Pilato, and D. Schicchi, "Deepeva: a deep neural network architecture for assessing sentence complexity in italian and english languages," Array, vol. 12, p. 100097, 2021.

[42] S. A. Crossley, S. Skalicky, M. Dascalu, D. S. McNamara, and K. Kyle, "Predicting text comprehension, processing, and familiarity in adult readers: New approaches to readability formulas," Discourse Processes, vol. 54, no. 5-6, pp. 340–359, 2017.

[43] M. Martinc, S. Pollak, and M. Robnik-Šikonja, "Supervised and unsupervised neural approaches to text readability," Computational Linguistics, vol. 47, no. 1, pp. 141–179, 2021.

[44] J. Liu and Y. Matsumoto, "Sentence complexity estimation for Chinese-speaking learners of Japanese," in Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation (R. E. Roxas, ed.), pp. 296–302, The National University (Phillippines), Nov. 2017.

[45] F. Liu and J. S. Lee, "Hybrid models for sentence readability assessment," in Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pp. 448–454, 2023.

[46] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," Psychological review, vol. 65 6, pp. 386–408, 1958.

[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, pp. 1735–80, 12 1997.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Proceedings

of the 31st International Conference on Neural Information Processing Systems, NIPS'17, (Red Hook, NY, USA), p. 6000–6010, Curran Associates Inc., 2017.

[49] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[50] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how bert works," 2020.

[51] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," 2023.

[52] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, T. T. Wang, S. Marks, C.-R. Segerie, M. Carroll, A. Peng, P. Christoffersen, M. Damani, S. Slocum, U. Anwar, A. Siththaranjan, M. Nadeau, E. J. Michaud, J. Pfau, D. Krasheninnikov, X. Chen, L. Langosco, P. Hase, E. Biyik, A. Dragan, D. Krueger, D. Sadigh, and D. Hadfield-Menell, "Open problems and fundamental limitations of reinforcement learning from human feedback," Transactions on Machine Learning Research, 2023. Survey Certification.

[53] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, (USA), p. 311–318, Association for Computational Linguistics, 2002.

[54] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, "Optimizing Statistical Machine Translation for Text Simplification," Transactions of the Association for Computational Linguistics, vol. 4, pp. 401–415, 12 2016.

[55] E. Sulem, O. Abend, and A. Rappoport, "Semantic structural evaluation for text simplification," in Proceedings of the 2018 Conference of the

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), (New Orleans, Louisiana), pp. 685–696, Association for Computational Linguistics, June 2018.

[56] P. Karisani and N. Karisani, "Semi-supervised text classification via self-pretraining," 2021.

[57] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018.

[58] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," 2020.

[59] S. Wu and M. Dredze, "Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT," 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), (Hong Kong, China), pp. 833–844, Association for Computational Linguistics, Nov. 2019.

[60] S. Štajner, K. C. Sheang, and H. Saggion, "Sentence simplification capabilities of transfer-based models," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 12172–12180, 2022.

[61] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 483–498, 2021.

[62] D. Ferrés, A. G. T. AbuRa'ed, and H. Saggion, "Spanish morphological generation with wide-coverage lexicons and decision trees," Procesamiento del Lenguaje Natural. 2017; 58: 109-116, 2017.

[63] Y. Huang, Y. Li, and Y. Luan, "Monolingual sentence matching for text simplification," arXiv preprint arXiv:1809.08703, 2018.

[64] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), (Vancouver, Canada), pp. 562–570, Association for Computational Linguistics, July 2017.

[65] X.-S. Hong, J.-J. Lee, S.-H. Wu, and M.-J. Jiang, "Cyut at the ntcir-16 finnum-3 task: Data resampling and data augmentation by generation," in In Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies, vol. 33, p. 95–102, Curran Associates, Inc., 2022.

[66] A. G. d'Sa, I. Illina, D. Fohr, D. Klakow, and D. Ruiter, "Exploring conditional language model based data augmentation approaches for hate speech classification," in International Conference on Text, Speech, and Dialogue, pp. 135–146, Springer, 2021.

[67] G. Sahu, P. Rodriguez, I. Laradji, P. Atighehchian, D. Vazquez, and D. Bahdanau, "Data augmentation for intent classification with off-the-shelf large language models," in Proceedings of the 4th Workshop on NLP for Conversational AI (B. Liu, A. Papangelis, S. Ultes, A. Rastogi, Y.-N. Chen, G. Spithourakis, E. Nouri, and W. Shi, eds.), (Dublin, Ireland), pp. 47–57, Association for Computational Linguistics, May 2022.

[68] F. Sufi, "Generative pre-trained transformer (gpt) in research: A systematic review on data augmentation," Information, vol. 15, no. 2, 2024.

[69] R. Watkins, "Guidance for researchers and peer-reviewers on the ethical use of large language models (llms) in scientific research workflows," AI and Ethics, pp. 1–6, 2023.

[70] J. E. Casal and M. Kessler, "Can linguists distinguish between chatgpt/ai and human writing?: A study of research ethics and academic publishing," Research Methods in Applied Linguistics, vol. 2, no. 3, p. 100068, 2023.

[71] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," ACM Comput. Surv., vol. 55, mar 2023.

[72] Z. Han, C. Gao, J. Liu, J. Zhang, and S. Q. Zhang, "Parameter-efficient fine-tuning for large models: A comprehensive survey," 2024.