UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LUCIANA REGINA BENCKE

Producing Synthetic Instances for Textual Classification and Natural Language Inference

Thesis presented in partial fulfillment of the requirements for the degree of Doctor of Computer Science

Advisor: Prof. Dr. Viviane Pereira Moreira

Porto Alegre
March 2024

CIP — CATALOGING-IN-PUBLICATION

Bencke, Luciana Regina

Producing Synthetic Instances for Textual Classification and Natural Language Inference / Luciana Regina Bencke. – Porto Alegre: PPGC da UFRGS, 2024.

110 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2024. Advisor: Viviane Pereira Moreira.

1. Natural language inference. 2. Entailment recognition. 3. Synthetic data. 4. Text generation. 5. Data augmentation. 6. Text classification. I. Moreira, Viviane Pereira. II. Título.

“The more I study, the more
insatiable do I feel my genius for it to be.”
— Ada Lovelace

AGRADECIMENTOS

Agradeço à família, amigos, colegas, professores e a todos que de alguma forma me ajudaram nesta jornada.

Aos meus amados pais, Nougathe e Elemar, pela educação que me foi dada e pelas lembranças de nosso convívio que me fortaleceram ao longo destes anos.

À minha amada madrinha Lordi (in memoriam), por me inspirar, através do seu exemplo de vida, a ser uma pessoa melhor.

Pai, Mãe e Lordi, obrigada pela vida que me foi dada e por todos os seus ensinamentos. Esse doutorado é uma homenagem a vocês.

Agradeço em especial ao meu amado Osvaldo, por toda a paciência, amor, dedicação, parceria e por sempre acreditar em mim, até quando eu mesma duvidei. Osvaldo, esse doutorado é dedicado a ti.

À minha orientadora, por estar sempre ao meu lado me animando diante dos problemas e também vibrando nos momentos de vitória, e que tanto me ensinou ao longo destes anos. Viviane você é um exemplo e uma inspiração.

Agradeço às colegas Moniele e Francielle por sua dedicação na revisão do dataset de NLI que é uma contribuição importante deste trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela bolsa de doutorado que me foi concedida.

À Universidade Federal de do Rio Grande do Sul, em especial ao Instituto de Informática, por possibilitar que pessoas se sintam valorizadas e motivadas a criar o futuro.

E, finalmente, agradeço a Deus e ao contínuo fluxo da vida que me surpreende, testa, mas também me emociona na busca por ser uma pessoa melhor e aproveitar essa chance ao máximo.

Obrigada!

Produzindo Instâncias Sintéticas para Classificação Textual e Inferência de Linguagem Natural

RESUMO

A tarefa de Inferência de Linguagem Natural (NLI) é um tipo especial de classificação de textos focada na dedução – um modelo é apresentado a um par de sentenças (premissa e hipótese) e classifica a relação entre os seus significados. Treinar modelos com conjuntos de dados para NLI é fundamental para sistemas semânticos. Além disso, conjuntos de dados de NLI são usados para treinar modelos de *sentence-transformers* (ST), que usam redes Siamesas para aprender a relação entre o par de sentenças, gerando boas representações (*embeddings*) em um espaço onde sentenças semelhantes ficam próximas. As *embeddings* de sentenças podem ser usadas como recursos para treinar outros modelos em tarefas como *clustering* e classificação. Os recursos existentes para NLI em português são limitados. Criar ou ampliar conjuntos de dados manualmente é custoso e requer conhecimento especializado. O aumento de dados (DA) oferece alternativas para superar essa limitação. DA é o primeiro passo para o desenvolvimento de instâncias sintéticas, e a geração de texto pode ser usada como um método de DA, especialmente ao utilizar o poder dos recentes grandes modelos de linguagens (LLM). Este trabalho se concentra na produção de um conjunto sintético de dados para NLI e na sua utilização para treinar modelos ST para gerar embeddings em português, empregando DA como primeiro passo para avaliar o comportamento da geração de texto. Com o objetivo de suprir a falta de recursos em português, esta tese apresenta o InferBR, um conjunto de dados sintéticos para NLI produzido empregando um processo majoritariamente automático. O InferBR foi utilizado para treinar modelos ST especializados em gerar embeddings em português, que apresentaram melhor desempenho que os modelos multilíngues existentes nas tarefas de *clustering*, classificação e similaridade semântica.

Palavras-chave: inferência de linguagem natural. reconhecimento de implicação. dados sintéticos. geração de texto. aumento de dados. classificação de texto.

ABSTRACT

Natural Language Inference (NLI) is a special type of text classification focused on deduction – a model is presented to a pair of sentences (premise and hypothesis) and classifies the relationship between their meanings. Training models with NLI datasets is key for semantic systems. NLI datasets are also used to train sentence-transformer (ST) models, which use Siamese networks to learn the relationship between the pair of sentences, generating good representations in an embedding space where similar sentences are placed close together. The sentence embeddings can be used as features to train other models for tasks such as clustering and classification. Existing NLI resources in Portuguese are limited. Creating or extending datasets manually is expensive and requires specialized knowledge. Data augmentation (DA) offers alternatives to overcome this issue. DA is the first step towards developing synthetic instances, and text generation can be used as a DA method, especially when utilizing the power of recent large language models (LLM). This work focuses on producing a synthetic NLI dataset and using it to train ST models for Portuguese embeddings, employing DA as the first step to evaluate the behavior of text generation. Aiming to cover the lack of resources in Portuguese, this thesis introduces InferBR, a synthetic NLI dataset produced using a mostly automatic process. InferBR was used to train ST models specialized in generating Portuguese embeddings, which presented better performance than the existing multilingual models in clustering, classification, and semantic similarity.

Keywords: natural language inference. entailment recognition. synthetic data. text generation. data augmentation. text classification.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BLEU	Bilingual Evaluation Understudy
BT	Back Translation
CBOW	Continuos Bag-of-Words
CKB	Commonsense Knowledge Bases
CNN	Convolutional Neural Networks
CWE	Contextual Word Embeddings
DA	Data Augmentation
EDA	Easy Data Augmentation
GPT	Generative Pre-Trained Transformer
ICL	In-Context Learning
ISO	International Organization for Standardization
LMPG	Language Model for Paraphrase Generation
LSTM	Long Short-Term Memory
MultiNLI	Multi-Genre Natural Language Inference
NLP	Natural Language Inference
NLP	Natural Language Processing
POS	Part-Of-Speech
RLHF	Reinforcement Learning with Human Feedback
RNN	Reccurent Neural Network
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
SNLI	Stanford Natural Language Inference
SST2	Stanford Sentiment Treebank (two classes)
ST	Sentence-Transformers
TG	Text generation

CONTENTS

1 INTRODUCTION

Text classification aims to assign labels to textual units such as words, sentences, paragraphs, or documents. It is a common task in Natural Language Processing (NLP) with applications in sentiment analysis, intent recognition, hate speech detection, document categorization, *etc.*

Natural Language Inference (NLI) classification, also known as Recognizing Textual Entailment (RTE), is a special case of text classification and presents additional challenges related to class boundaries, which may be fuzzy and ambiguous, sometimes requiring world knowledge. It can be seen as a classification task focused on deduction (SALVATORE et al., 2023) – a model is presented with a pair of sentences and classifies the relationship between their meanings (JURAFSKY; MARTIN, 2023). The first sentence is known as the *premise* (P), and the second is the *hypothesis* (H). An NLI model should infer whether (i) H entails P (*i.e.,* based on P, we can infer H is true), (ii) H contradicts P (*i.e.,* based on P, we can infer H is false), or (iii) H is neutral in relation to P (the truth of H cannot be determined on the basis of P).

Understanding the relationships in NLI is fundamental because their semantic concepts are central to all aspects of natural language meaning (BOWMAN et al., 2015a; BENTHEM, 2008; KATZ, 1972). Training good models for NLI is key for semantic systems (MARELLI et al., 2014). NLI datasets help Question-Answering (QA) determine whether a given text passage contains the answer to a specific question by analyzing the entailment relationships between the question and the text passage, identifying relevant information, and providing better answers (CHEN; CHOI; DURRETT, 2021; WANG et al., 2019). Similarly, an NLI model can be applied to standard text classification because it learns the logical relationship between pairs of sentences. The sentence to be classified can be arranged as the premise and the label as the hypothesis.

Learning to classify how H relates to P is useful for the development of semantic representations (BOWMAN et al., 2015a). Devlin et al. (2019) showed the power of the embedding representations generated by models like BERT. The authors elucidated two strategies for using pre-trained language model representations in downstream tasks: feature-based and fine-tuning. The feature-based technique uses task-specific architectures that include the pre-trained representations as features, while fine-tuning introduces minimal task-specific parameters and is trained on the downstream tasks by fine-tuning all or part of the pre-trained parameters. Siamese networks (BROMLEY et al., 1993)

have been used to train Sentence-Transformers (ST) models (REIMERS; GUREVYCH, 2019) with NLI datasets. The resulting models produce better embeddings than encoder models like BERT and are faster to execute similarity search (REIMERS; GUREVYCH, 2019; GAO; YAO; CHEN, 2021). Basically, they encode embeddings for sentences and paragraphs, which can then be fed into a task-specific architecture as features.

NLI datasets are valuable to train ST models since learning the relationship between a pair of sentences helps generate an embedding space where similar sentences are close while dissimilar are apart. There are several models available to provide sentence embeddings. Portuguese (and other languages different from English) used multilingual models (REIMERS; GUREVYCH, 2020) that were created employing knowledge distillation techniques (HINTON; VINYALS; DEAN, 2015), in which the teacher is the English model trained on large datasets. The student model distills the teacher's knowledge, learning a multilingual sentence embedding space assuming that vector spaces are aligned across languages. To improve embedding quality for a specific language or domain, one can fine-tune the multilingual model with NLI datasets in that language/domain. ST models have also been used for few-shot learning: fine-tuning an existing ST model to a small classification dataset in a contrastive Siamese manner (TUNSTALL et al., 2022). The resulting model is then used to generate the text embeddings for the instances in the small dataset, which are then used to train a classification head, such as a simple logistic regression.

According to Sadat and Caragea (2022), creating new NLI datasets capturing linguistic properties of different domains is complex. Efforts towards reducing the reliance on manually annotated data in training deep learning models for NLI are welcome. We identified only two NLI datasets publicly available for Portuguese: ASSIN2 (REAL; FONSECA; OLIVEIRA, 2020) and SICK-BR (REAL et al., 2018). The former has only two classes, entailment and non-entailment, and it was built using sentences from the latter, which has the three NLI classes.

Producing manually labeled datasets is very expensive; it requires specialized knowledge and demands time to annotate adequately sufficient data. In real-world settings, a few labeled instances are sometimes the only resource available to train classifiers. The lack of labeled data can be a consequence of the limited access to experts, absence of high-variance data, or environments that tend to generate large volumes of data, making manual label assignment unfeasible (PALEYES; URMA; LAWRENCE, 2020). Sometimes, the effort to get more data is not feasible, for example, when there is a very rare

class, as can happen in the medical domain that also suffers from data restrictions related to privacy concerns and labeling costs (GARCEA et al., 2022). Beyond this, some languages still do not get enough attention from NLP researchers. As a result, there is a lack of labeled datasets for several tasks. They are called low-resource languages. Specifically for NLI, the difficulty is not only labeling instances but building good pairs of premises with hypotheses that can be used to train an ST model to generate useful representations.

The state-of-the-art in text classification, which includes NLI, is the same as in several NLP tasks and involves fine-tuning Large Language Models (LLM) based on Transformers (VASWANI et al., 2017). Large models trained on small datasets tend to overfit and thus produce inaccurate results (SHORTEN; KHOSHGOFTAAR; FURHT, 2021; CHEN et al., 2023). Small datasets may also cause models to underfit since data is insufficient for learning. Data augmentation (DA) techniques aim to tackle this problem. DA has been used to synthetically inflate data for training to obtain models with greater generalization power (PELLICER; FERREIRA; COSTA, 2023).

DA encompasses methods of increasing training data variety without literally collecting more data. These methods are based on strategies that either add modified copies of existing data or create synthetic data, aiming for the augmented data to act as a regularizer and reduce overfitting when training machine learning models (FENG et al., 2021). Text generation using LLM can be employed as a DA strategy to generate synthetic instances for classification tasks (YOO et al., 2021; BAYER; FREY; REUTER, 2023). DA enables the generation of synthetic labeled instances based on the original data by modifying the features with transformations that do not change the label assigned to the base instance (HEDDERICH et al., 2021). For example, in a sentiment analysis setting, consider the sentence "*one of those exceedingly rare films in which the talk alone is enough to keep us involved*". The sentence was annotated with a Positive class label. It could be augmented by generating the following similar sentences: i)"*the talk alone is enough to keep us connected to one of those unusual films*"; ii) "*the talk itself is sufficient to involve us in a film like this*". The generated sentences respect the semantic limits in such a way that the label is preserved.

With recent unprecedented advances in Large Language Models (LLM) highly specialized in following human instructions (OUYANG et al., 2022), text generation established itself as a viable option to generate synthetic data for several tasks. Gartner[1], a company well known for delivering a broad spectrum of IT-related research, in-

[1]https://www.gartner.com/

sights, and guidance for developing strategies and choosing technologies, has predicted that 60% of the data used for training Artificial Intelligence (AI) models in 2024 will be synthetic (KEEN, 2023). In 2021, the company estimated this number was up from 1%.

Nikolenko (2021) underlines that data augmentation can be seen as the first step towards synthetic data: "there is no synthetic data generation, but there is recombination and adaptation of existing real data, and the resulting instances often look quite 'synthetic'". The author remarks that in NLP, augmented data is similar to synthetic data: "to expand a dataset, one can replace words with their synonyms, getting 'synthetic sentences' that can still preserve target variables such as the topic of the text, its sentiment, and so on". In this work, we use the term "synthetic" also for instances resulting from augmentation techniques that modify real data.

We hypothesize that *generating text using recent generative LLM may succeed as an augmentation method for Portuguese presenting potential and flexibility that allows us to achieve our main goal: to develop a synthetic NLI dataset for Portuguese and use it to train an ST model for that language*.

The difficulty in generating such synthetic instances for NLI is not only related to *how* to generate such pairs but also if they would be correctly labeled respecting the entailment, contradiction, and neutral concept boundaries. The challenge is to explore semantic variations and still maintain the correct classes.

1.1 Research Questions

Given the context above, this thesis aims to answer the following questions:

RQ1. Which augmentation strategies should be selected for text classification to pave the understanding of generative approaches to produce synthetic instances?

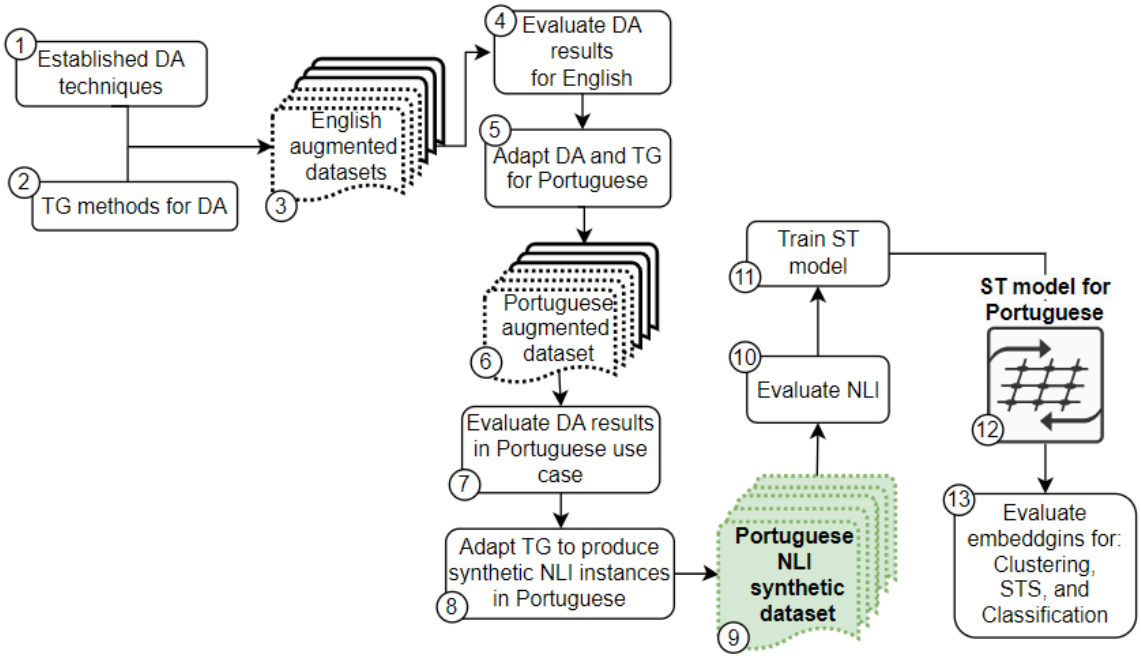
RQ2. Besides improving the final classification results, what else is important in DA to ensure high-quality synthetic instances?

RQ3. Is text generation competitive for DA? What are the pros/cons compared to other methods? What insights acquired exploring DA can be investigated in the next step to produce synthetic instances for NLI in Portuguese?

RQ4. How to generate a synthetic dataset for NLI in Portuguese? How to evaluate the results?

RQ5. How to use the NLI dataset to train a sentence embedding model for Portuguese?

Figure 1.1 – Macro-processes to generate the NLI dataset and ST model



Source: the author

How do we evaluate results?

1.2 Solution Overview

The process followed in this work to answer the research question is described in Figure 1.1. First, we investigated augmentation strategies for text classification using established DA methods and techniques based on text generation (TG), corresponding to steps (1) and (2). We assembled low-data scenarios in English (3) and evaluated the classification results on the augmented datasets (4). We adapted some of the used DA methods in English to Portuguese (5), including TG, and applied them to a use case (6). We quantitatively and qualitatively evaluated classification results and the synthetic instances in the Portuguese augmented dataset (7). We adjusted the TG method used in DA to generate premises and hypotheses for NLI in Portuguese (8). Once the NLI dataset was produced (9) and evaluated (10), we used its pairs (premise and hypothesis) to train a Sentence-Transformers (ST) model (11). The quality of the embeddings generated by this model (12) was evaluated in Portuguese datasets for clustering, semantic similarity, and classification (13).

1.3 Contributions

The contributions of this thesis are:

C1. An analysis of the impact of DA methods over simulated low-data scenarios using well-known public text classification datasets in English. This contribution is described in Section 4.1.

C2. A dataset named City-tweets with 1993 tweets in Portuguese labeled with the Smart City dimension they refer to. The dataset is introduced in Section 4.2.1.

C3. The adaptation of DA techniques that were originally designed for English classification tasks to work with Portuguese, described in Section 4.2.2.

C4. A comparison of the DA techniques in a multiclass single-label classification problem with a Portuguese dataset both in quantitative and qualitative terms. The results of this comparison are shown in Sections 4.2.3 and 4.2.4.

C5. A synthetic dataset for NLI in Portuguese, indicating the quality of the instances after human validation. The InferBR dataset is detailed in Section 5.3.1.

C6. The processes for generating NLI datasets that may also be adapted for tasks other than NLI and be reused by researchers working on low-resource languages. The processes are described in Sections 5.1 and 5.2.

C7. A model to generate embeddings for Portuguese sentences. The model is described in Section 6.4.

We published part of this work related to contributions C1 to C4 in the Language Resources and Evaluation journal, which can be found in Bencke and Moreira (2023). The contributions C5 and C6 were accepted for publication at the International Conference on Language Resources, Evaluation, and Computational Linguistics (LREC-COLING 2024). We are working on a paper for C7 and make the model available.

1.4 Structure of the Text

The remainder of this work is organized as follows. Chapter 2 presents concepts foundational to this thesis or related to the proposed methods. Related works are described in Chapter 3 covering works using DA and synthetic instances for text classification and NLI research, highlighting important human-generated datasets and works that produced

synthetic instances for NLI. Chapter 4 describes the DA analysis over the English datasets in simulated low-data regimes and the adaptation of the methods to Portuguese. In this later task, we assembled a small dataset and applied the adapted methods to it, evaluating the results. In Chapter 5, we used the knowledge acquired when applying DA with text generation methods to generate synthetic premises and hypotheses and create a new NLI dataset in Portuguese, InferBR. Still, in this chapter, we described the human evaluation and the comparison to models trained with the other two existing datasets in Portuguese. In Chapter 6, we present the results of training the ST model with NLI data in Portuguese. Finally, in Chapter 7, we summarize this work and discuss the limitations, ethical concerns, and opportunities for future work.

2 BACKGROUND

This chapter covers the foundational concepts studied to develop this thesis and the historical evolution of techniques and models used in the experiments. In this work, we use state-of-the-art language models in several steps. Section 2.1 addresses the evolution of language models from the first statistical models to the recent deep learning architectures named Transformers (VASWANI et al., 2017) and the spread use of Transfer Learning techniques such as fine-tuning. Since we used LLM to generate text and create synthetic instances, we describe recent text generation approaches in Section 2.2. In Sections 2.4 and 2.3, we also provide background on data augmentation and supervised text classification because we are using DA for text classification as the first step to evaluate text generation in producing synthetic instances.

2.1 Language Models

Models that assign probabilities to sequences of words are called language models. These models can then predict a word from preceding words (JURAFSKY; MARTIN, 2023). There are historically different ways of estimating this probability distribution. Those are described in the next subsections.

2.1.1 Statistical Language Models

Before the consolidation of deep neural networks in the NLP field, language models were built using statistical approaches. Examples of those models are n-gram and hidden Markov models (MANNING; SCHUTZE, 1999). Statistical language models suffer from a fundamental problem that makes language modeling difficult – the curse of dimensionality: if one wants to model the joint distribution of 10 consecutive words in a natural language with a vocabulary V of size 100,000, there are potential $100,000^{10}-1 = 10^{50}-1$ free parameters (BENGIO; DUCHARME; VINCENT, 2000). This can be understood in Equation 2.1 where w_t is the t-th word on the sequence T:

$$P(w_{t=1}^{T}) = \prod_{t=1}^{T} P(w_t|w_1^{t-1}) \tag{2.1}$$

To deal with the curse of dimensionality, n-gram models make the approximation $P(w_t|w_1^{t-1}) \approx P(w_t|w_{t-n+1}^{t-1})$ and calculate conditional probabilities for the next word considering the last $n-1$ words, as follows in Equation 2.2.

$$P(w_{t=1}^{T}) = \prod_{t=1}^{T} P(w_t|w_{t-n+1}^{t-1}) \tag{2.2}$$

Thus, n-gram models have limits in terms of scalability and generalization: the number of parameters increases exponentially as the n-gram order increases, and small n may compromise the model to generalize from training to test set (JURAFSKY; MARTIN, 2023).

2.1.2 Neural Language Models

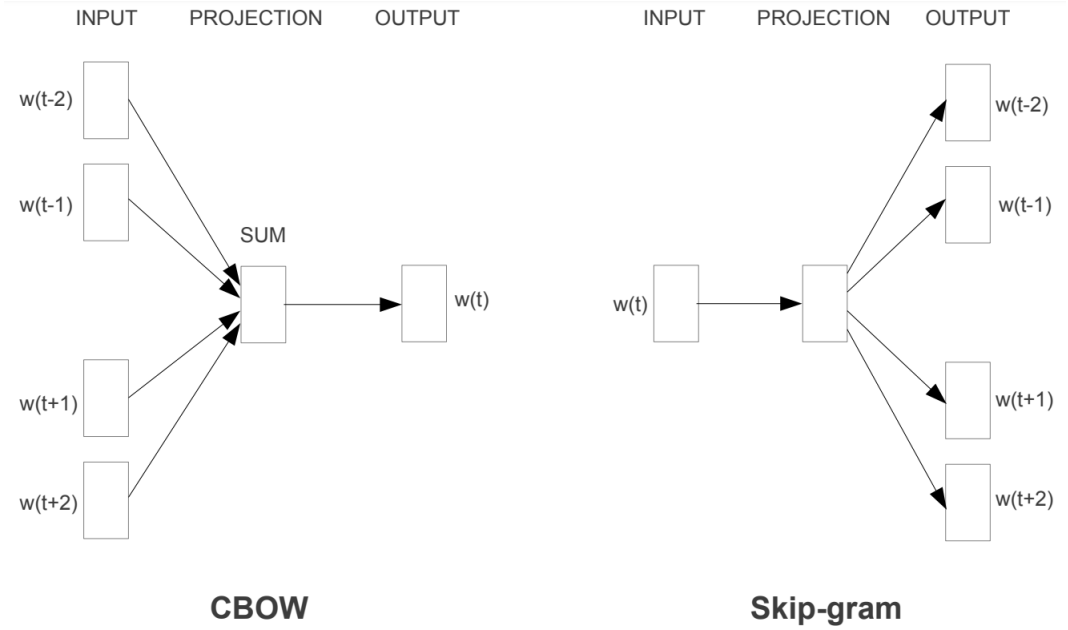
Compared to n-gram models, neural language models can handle much longer text and can generalize better over contexts of similar words, achieving better results predicting words. Nevertheless, neural network language models are much more complex, difficult to interpret, and require more energy to train (JURAFSKY; MARTIN, 2023)

Bengio, Ducharme and Vincent (2000) introduced a new way of modeling language using neural networks where the goal of the model was to learn (1) a distributed representation for each word (*i.e.,* a similarity between words) along with (2) the probability function for word sequences expressed with these representations. Authors remarked that generalization was obtained because a sequence of words that has never been seen before had a high probability of being composed of words similar to words forming an already seen sentence. They used a feedforward neural network to predict the next word, given the previous n words, outperforming traditional n-gram models. They demonstrated that neural language models could also be used to develop embeddings for the word prediction task (JURAFSKY; MARTIN, 2023). Their work was foundational in developing word embeddings, where words are represented as vectors in a continuous vector space, and words with similar contexts have similar representations.

2.1.3 Word Embeddings

Mikolov et al. (2011) demonstrated that recurrent neural networks (RNN) could be used as language models. Authors further simplified the hidden layers of these net-

Figure 2.1 – Word2vec framework

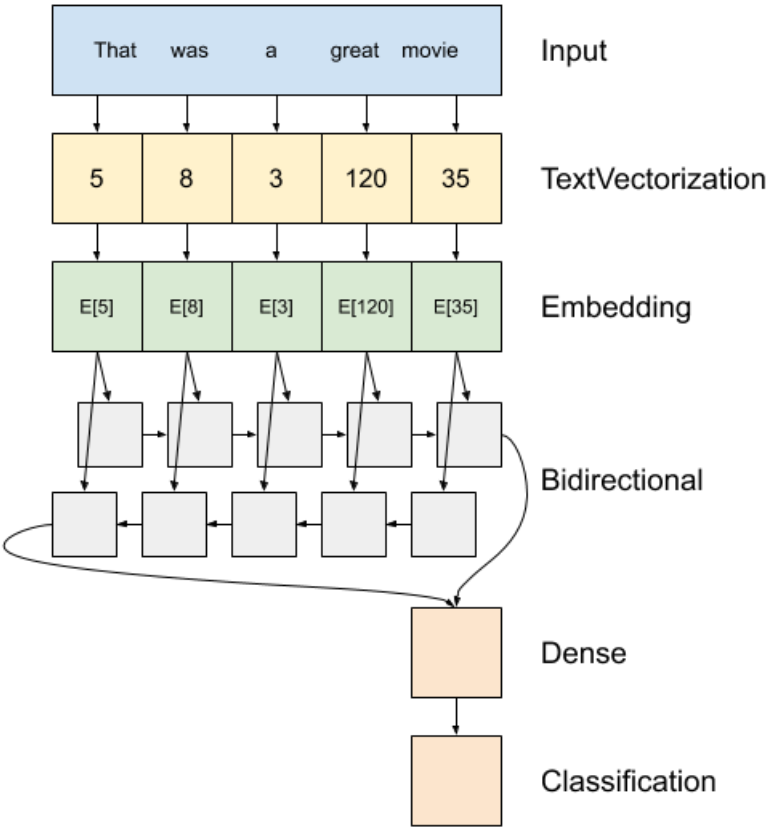


Source: Mikolov et al. (2013)

works, using shadow models, the Skip-gram and CBOW architectures from Word2Vec framework (MIKOLOV et al., 2013). They also proposed the negative sampling training algorithm (MIKOLOV et al., 2013). The main idea is to use a shallow neural network to learn word embeddings by predicting a word's context (in the case of Skip-gram) or a context word given a word (in the case of CBOW), as presented in Figure 2.1. This process results in word vectors that capture semantic and syntactic relationships between words. Other word representations were also developed, such as Glove (PENNINGTON; SOCHER; MANNING, 2014) and FastText (BOJANOWSKI et al., 2017).

The word embeddings static representations learned by the aforementioned works can be used in other neural networks. Figure 2.2 shows an abstraction of how to use a layer containing the embeddings of a specific input. The first layer is the encoder, which converts the text to a sequence of token indices (TextVectorization). Each word (now represented by its index in the vocabulary) is then converted to a vector of embeddings (that can be extracted from embeddings files containing word vectors of words trained with large corpora, or it can train the embeddings while adjusting the network considering the existing corpus). Those embeddings are fed into the network structure that, in the example, is composed of a bidirectional LSTM (a kind of RNN) that processes sequence input by iterating through the elements and passing the outputs from one timestep to their input on the next timestep. The bidirectional characteristic ensures the propagation of

Figure 2.2 – Example of the use of embeddings to train a text classifier



Source: TensorFlow Tutorials (2023)

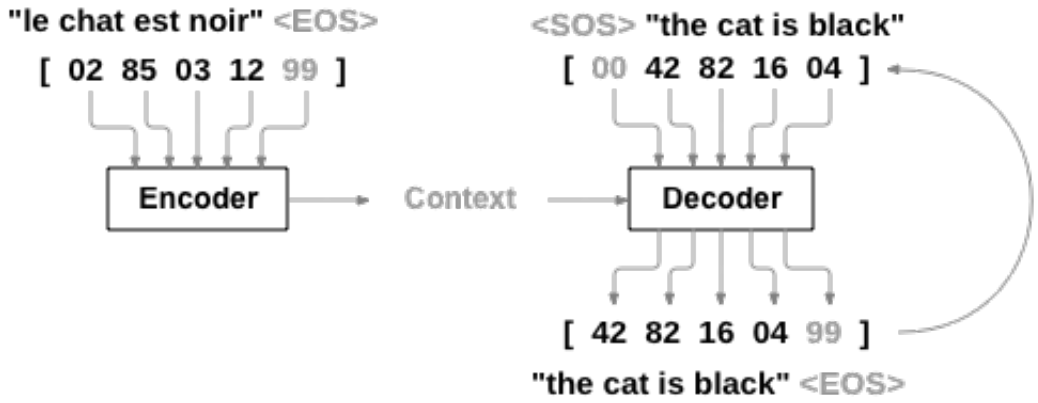
the input forward and backward through the LSTM layer and then concatenates the final output. In the end, the LSTM had converted the input sequence into a single vector and passed it to the two next final dense layers, which do the final processing and convert the representation vector to a classification output.

2.1.4 Sequence-to-Sequence Models

Seq2seq models are designed to transform a sequence from a source domain to a sequence in a target domain, where both sentences may have different lengths. In NLP, these models were first applied in machine translation (SUTSKEVER; VINYALS; LE, 2014) but are widely used in text summarization and question-answering.

The seq2seq architecture typically consists of two main components: the encoder and the decoder. The encoder usually uses RNNs, or variations named LSTM or GRUs, which are RNNs more capable of dealing with longer sentences. The encoder processes each element of the input sequence one by one, updating its internal state. When it reaches

Figure 2.3 – Example of Sequence-to-Sequence models



Source: PyTorch Tutorials (2023)

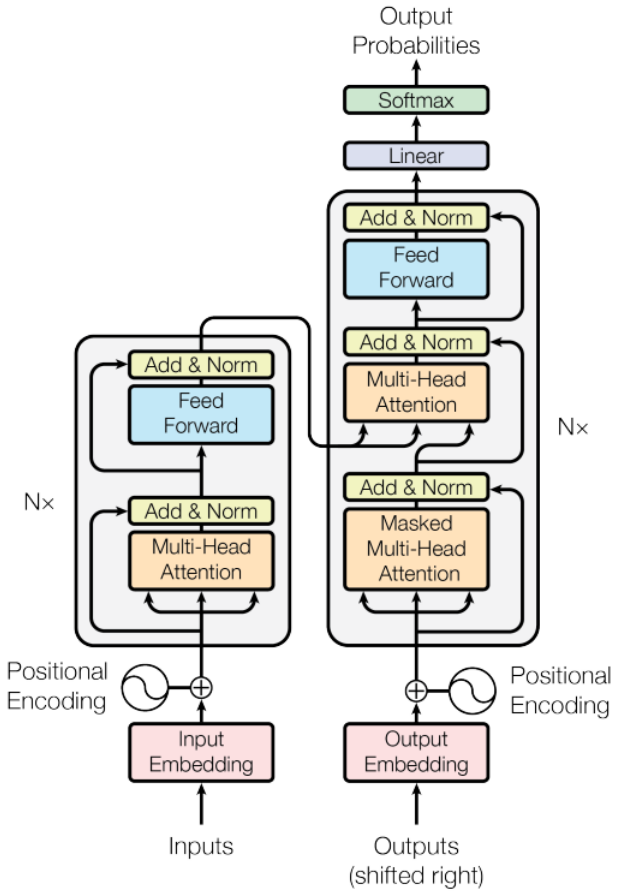
the end-of-sentence token <EOS>, the decoder's internal state represents the entire input sequence: the context vector, as depicted in Figure 2.3. The decoder takes the context vector and generates the output sequence element by element. Like the encoder, it is typically an RNN, LSTM, or GRU network. It begins with a special start-of-sequence <SOS> token and generates the next token based on the context vector and the previously generated tokens, continuing this process until an <EOS> token or a maximum length is reached.

Bahdanau, Cho and Bengio (2015) developed attention mechanisms in seq2seq models, allowing the decoder to focus on different parts of the input sequence in each step of the output generation. Authors highlight that by letting the decoder have an attention mechanism, they relieved the encoder from the burden of having to encode all information in the source sentence into a fixed-length vector. It was a significant advancement, especially for translating longer sentences.

2.1.5 Transformers

The success of seq2seq models and attention mechanisms directly influenced the development of transformers (VASWANI et al., 2017), which revolutionized the NLP field. Transformers have encoder and decoder stacks, as shown in Figure 2.4, where N_x corresponds to the number of encoders or decoders in each stack. In the original paper, the encoder and decoder stacks have both $N = 6$ identical layers. Each layer has two

Figure 2.4 – The Transformer architecture



Source: Vaswani et al. (2017)

sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. The decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Positional Encodings are used to keep the information about tokens in order.

Transformers are entirely based on attention mechanisms and do not use recurrence or convolution. Thus, they are highly parallelizable, making them more scalable and efficient for training on large datasets. They can also handle long-range dependencies in the data more effectively than LSTMs.

There are several architectural variations of transformers: there are some based only on the encoder or the decoder and those that use both, but they can also have different numbers of layers, changes in attention mechanisms, feed-forward networks, positional encodings, and more. We generally classify the transformers as encoder-only, decoder-only, and encoder-decoder. Some influential models are described next.

- BERT: for Bidirectional Encoder Representations from Transformers (DEVLIN et

al., 2019). It is an encoder-only that was pre-trained using the masked language model (MLM) approach where random tokens are masked, and the model learns to predict them based on context, along with next sentence prediction (NSP), in the same way as MLM, the model tries to predict the next sentence. BERT family models are applied in text classification, question answering, named entity recognition, and other tasks that require understanding the context of each word.

- GPT: for Generative Pre-trained Transformer (RADFORD et al., 2018). It is a decoder-only model pre-trained using an unsupervised approach where the model predicts the next token in a sequence. There are several sizes of modes from GPT-1 to GPT-4, and for those that got details published, the size ranges from 117 million in GPT (original) to 175 billion in GPT-3. They are applied in text generation, conversational agents, and any task involving generating coherent text sequences.
- BART: for Bidirectional and Auto-Regressive Transformers (LEWIS et al., 2020). It is a full encoder-decoder that combines bidirectional conditioning (like BERT) and autoregressive conditioning (like GPT). The training is based on corrupting text with an arbitrary noising function, and the model learns how to reconstruct it. The model has 139 million in BART-base and 406 million in BART-large. The model can be applied in text generation (e.g., summarization, translation) but also works well for comprehension tasks (e.g., text classification, question answering).
- T5: for Text-to-Text Transfer Transformer (RAFFEL et al., 2020). It is an encoder-decoder architecture trained on a multi-task objective, converting every NLP problem into a text-to-text format where input and output are text strings. Model size ranges from 60 million to 11 billion parameters across various model sizes. T5 models are used in summarization, translation, question answering, and classification tasks, among others.

Nonetheless, Transformers are traditionally memory-intensive, especially for longer sequences, due to the self-attention mechanism, which has a complexity of $O(n^2)$: for each token in the input sequence, it computes attention scores with every other token. Thus, models have a token limit beyond which they cannot process input in a single batch, e.g., 512 tokens for BERT, 1024 for GPT-2, 2,048 for GPT-3, 4,096 for GPT-3.5 and more recently, GPT-4 beta version with models with from 8k to 34k. In addition to these computational complexities, the trend towards larger models (with more parameters) in pursuit of state-of-the-art performance also demands substantial memory during inference.

2.1.6 Transfer Learning

Transfer learning in machine learning refers to acquiring knowledge from one task or domain and then applying it (transferring it) to solve a new task (JURAFSKY; MARTIN, 2023). Fine-tuning is one kind of transfer learning that takes advantage of the knowledge in these models like BERT, GPT, etc. We can fine-tune a model by adding a classifier head that takes the top layer of the model as input to perform downstream tasks like named entity tagging, question answering, text classification, etc. (JURAFSKY; MARTIN, 2023).

The fine-tuning process uses a much smaller dataset that contains the labels needed for the downstream task the models will be adjusted for. The process can update the entire pre-trained model during training on a downstream task or have part of the weights or all of them frozen.

2.2 Text Generation

In the early 1950s, Alan Turing proposed the well-known Turin Test (TURING, 1950), which set the stage for evaluating machine-generated text. Today, we discuss if this test is still valid since it is an imitation test and recent models are highly capable of this (The Conversation, 2023). At that time, text was generated using rule-based systems. ELIZA (WEIZENBAUM, 1966) was a pioneer dialog system[1] that used pattern-matching substitution methodology to simulate a psychotherapist, and, despite some out-of-context answers, many people believed they were talking to a human.

Text generation (TG) in the context of the LLM, introduced in the previous section, refers to producing sequences of text conditioned on an input text (LIU et al., 2023; ZHANG et al., 2022). It is used in several NLP tasks, such as summarization, translation, open-ended text generation, speech-to-text, vision-to-text, *etc.*. According to Jurafsky and Martin (2023), using a language model to generate text is where the impact of neural language models on NLP has been the largest. Text generation, along with image generation and code generation, constitute a new area of AI that is often called generative AI.

Autoregressive language models such as GPT2 (RADFORD et al., 2019) and GPT-3 (BROWN et al., 2020) have extensively been used in text generation tasks. These models are trained on Causal Language Modeling (ZHU et al., 2023), *i.e.,* each gener-

[1] Dialog systems rely on text generation techniques to create responses in conversations.

ated token considers the context of the previous words. The autoregressive characteristic refers to how these models work: after each token is produced, that token is added to the sequence of inputs, and that, in turn, becomes the input to the model in its next step (ALAMMAR, 2019).

2.2.1 Decoding Strategies

Language models calculate the probability that a word in their vocabulary would come after a given input sequence. Decoding strategies are methods used to convert the output probabilities provided by a language model into an appropriate sequence of words. Different decoding strategies can lead to significantly different results regarding text quality, coherence, and diversity. Some of the most commonly used decoding strategies are the following.

- Greedy Search: it selects the word with the highest probability as its next word. It is the simplest method but computationally very efficient. It provides low diversity, leading to repetitions. A greedy algorithm makes a locally optimal choice, regardless of whether it will be the best choice with hindsight (JURAFSKY; MARTIN, 2023).
- Beam Search (FREITAG; AL-ONAIZAN, 2017): it tracks a fixed number of hypotheses (number of beams) at each step and chooses the best overall sequence at the end, considering the cumulative probability of each selected sequence. It is computationally expensive and still prone to repetition.
- Top-k Sampling (FAN; LEWIS; DAUPHIN, 2018): chooses randomly the top k most likely next words, where k is a parameter. The model randomly picks the next word according to a probability distribution limited to the k most likely next word. It introduces randomness, generating more diverse outputs. It may generate less coherent text, especially if k is high.
- Top-p Sampling (HOLTZMAN et al., 2019): it chooses the smallest possible set of words whose cumulative probability exceeds the probability p. The probability mass is then redistributed among this set of words. The value of p controls how diverse the generated text is. Higher values of p lead to more diverse text, while lower values generate more predictable text.
- Temperature: lowering the softmax temperature means making the probability dis-

tribution sharper (KAMATH; LIU; WHITAKER, 2019). Temperature is a hyperparameter that controls the randomness of the decoding process in LLM. Lower values result in more predictable and repetitive text, and the model becomes deterministic, and a higher temperature generates more diverse and creative text.

The choice of a decoding strategy has to deal with a trade-off between diversity, coherence, and computational efficiency. In many cases, a combination of these strategies or custom modifications can be used to balance them.

2.2.2 In-Context Learning

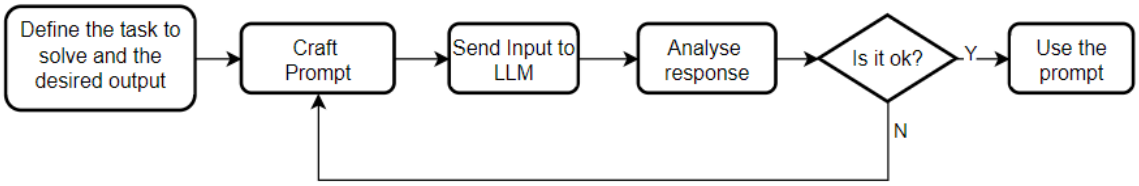
In-context learning (ICL) refers to the ability of large models to adapt to a task based on the context provided in the input without any parameter updates (BROWN et al., 2020) at inference time. The model generalizes promptly from a few examples of a new concept not previously trained without any gradient updates to the model (CHAN et al., 2022).

ICL can happen in a zero-shot setting, *i.e.,* only the request (the task to be solved) is sent to the model with the test instance for which the model should make a prediction. On the other hand, few-shot in-context learning refers to using a set of demonstrations on the target task consisting of input and desired output. So, besides sending the task description, we sent the demonstrations, followed by a single unlabeled example for which the prediction is desired. The model should benefit from those examples to provide the answer to the task it was not specifically trained on. All this input is called "prompt".

Models capable of in-context learning are typically pre-trained on diverse and extensive datasets, encompassing multiple tasks. The way this learning exactly happens is an open question, with some recent works attempting to explain this capability (ZHANG; ZHOU; LIU, 2023; WEI et al., 2023; OSWALD et al., 2023; GARG et al., 2022). Prompt engineering is the practice of designing and refining input prompts to improve the responses we obtain from the language model. It is an empirical activity. The outputs of the model vary according to the prompt used, the examples selected, and the order of the examples. Figure 2.5 shows a process of manual crafting prompts. It is also possible to reuse the LLM responses to the prompts, developing a chain of actions based on the LLM responses.

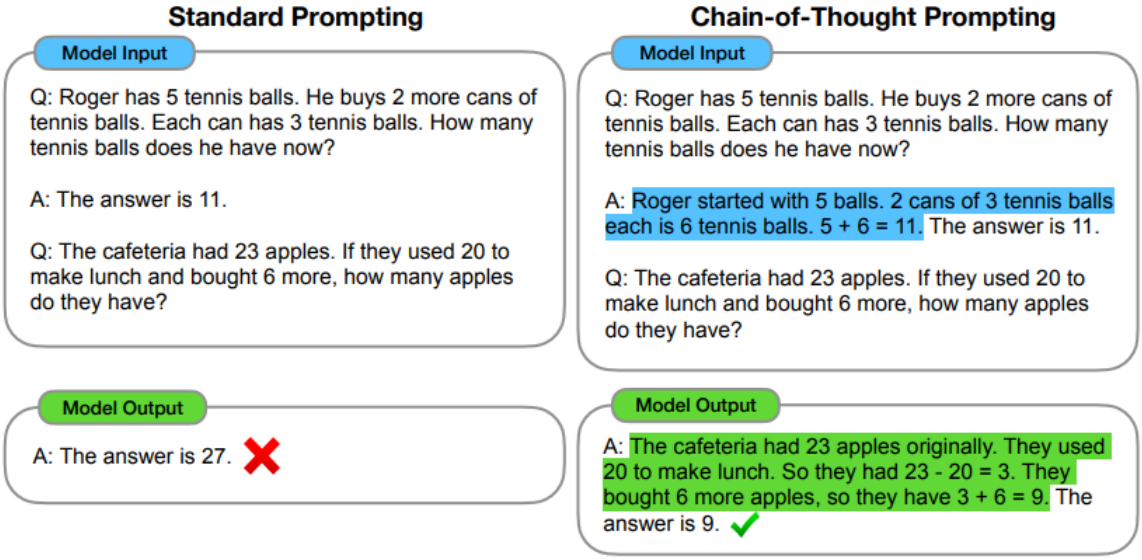
The second step in Figure 2.5 (craft prompt) can be automated and/or use external

Figure 2.5 – Prompt Engineering Loop



Source: the author

Figure 2.6 – Chain-of-Thought Prompting for Reasoning in LLM



Source: Wei et al. (2022)

sources to complement the prompt and give more relevant context to improve LLM response. Several prompt creation strategies have been developed. One of the most used is the Chain of Thoughts (WEI et al., 2022) presented in Figure 2.6. It enhances the ability to perform intricate reasoning by incorporating intermediary thinking steps. This method can be integrated with few-shot learning.

2.2.3 Instruction Tuning

The recent development of LLM has shown a high increase in the size of those models. According to (OUYANG et al., 2022), making language models bigger does not make them better at following users' intent. Even big models generate outputs that are untruthful, toxic, or just not helpful to the user. Due to these drawbacks, it is common to

say these models are not aligned with their users' intentions.

The final goal of Instruction Tuning is to align model responses with human expectations. According to Lai et al. (2023), there are two major approaches for Instruction Tuning in NLP: Reinforcement Learning with Human Feedback (RLHF) and the supervised fine-tuning approach. In the latter, the pre-trained LLMs are fine-tuned over the instruction triples (instruction, input, output) via supervised learning to promote their alignment with human expectations.

Reinforcement learning is a paradigm of machine learning where an agent (a robot, for example) learns to take actions (move inside a maze looking for the exit) in order to maximize a reward (a positive reward is received when it moves close to the exit). Reinforcement learning with human feedback (RLHF) is a technique that combines reinforcement learning approaches with human guidance to train a NLP agent (a language model or a chatbot, for example). The main difference lies in the reward function and the source of the reward signal. Traditional RL often relies on rewards provided by the environment, while RLHF uses the feedback provided by humans.

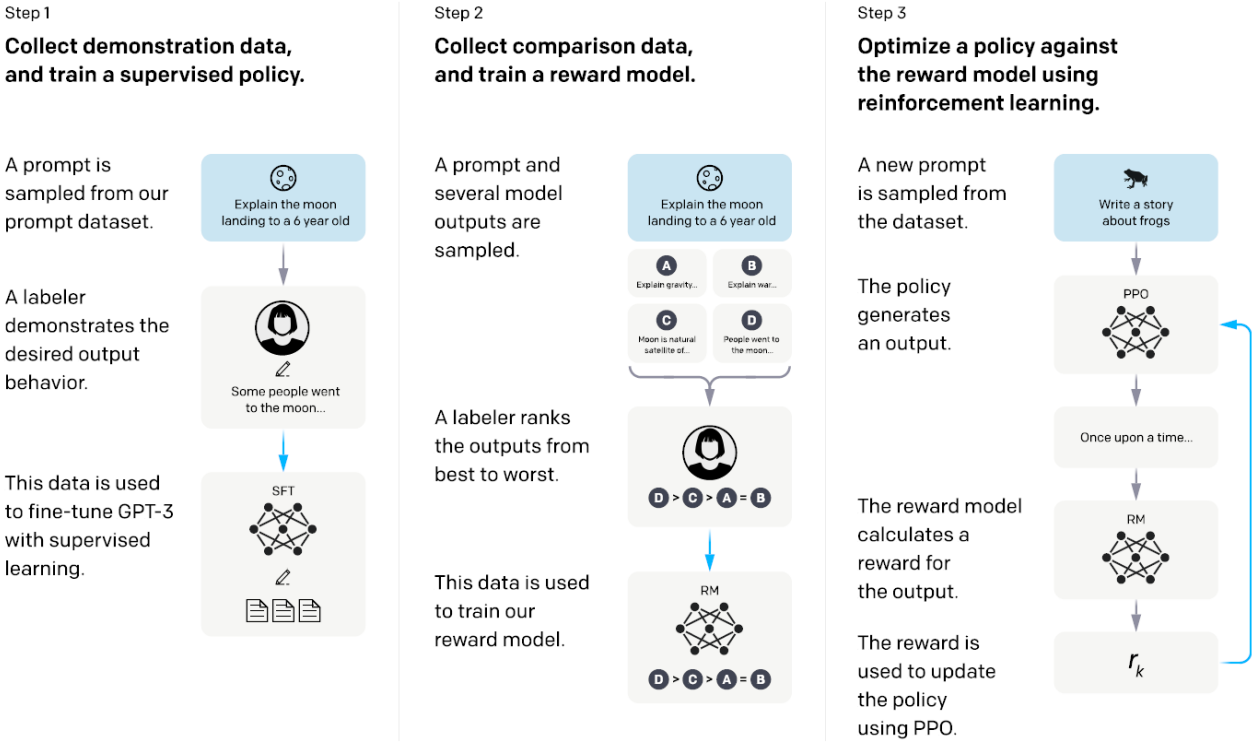
RLHF was used to optimize GPT-3 (OUYANG et al., 2022), creating a series of models called *InstructGPT* (GPT-3.5). The overview of the three-step process involved in their training approach is presented in Figure 2.7. Authors report that in human evaluations, outputs from the *InstructGPT* models are preferred to outputs from the GPT-3, despite it having 100 times fewer parameters.

2.2.4 OpenAI API

Currently, there has been an avalanche of LLMs. For the text generation tasks, we chose the GPT model family from OpenAI. The research path from GPT (RADFORD et al., 2018), GPT-2 (RADFORD et al., 2019), and GPT-3 (BROWN et al., 2020) and GPT-4 (OPENAI, 2023) leverages more data and computation, which, as advocated by OpenAI, is necessary for the development of increasingly sophisticated and capable language models. The first two models were open, but since GPT3, the company made the model available through an API and has not released the trained model weights as open-source. The company justified it on the safety risk due to the possibility of model misuse, and they want to make it available in a controlled environment. The API revenue would support future research (OpenAI, 2023).

GPT-3 has improved considerable text generation capabilities, and afterward, the

Figure 2.7 – Overview of the RLHF used by *InstructGPT* models



Source: Ouyang et al. (2022)

InstructGPT (GPT-3.5) models trained with RLHF improved the capacity to meet user intent. The company also launched models optimized for chatting: `GPT-3.5-turbo` and `GPT-4` (OPENAI, 2023). Both are conversation models that accept a sequence of messages as input and produce a message as the output. While the chat-based structure is intended to facilitate multi-turn dialogues, it is equally effective for single-turn tasks that do not involve any conversation (similar to the tasks handled by instruction-following models like `text-davinci-003`[2]) (OPENAI, 2023). GPT-4 is a multimodal model with improvements compared to GPT-3.5 models, especially in factuality and adherence to the desired behavior.

2.3 Supervised Text Classification

Classification involves predicting the category for new instances based on a model derived from data used in training. Text classification acts over a text unit, which can be a token, a sentence, a paragraph, or a whole document. In multiclass classification, every

[2]<https://platform.openai.com/docs/models>

instance belongs to only one class (when an instance can have more labels, it is named multilabel classification), and there are more than two classes (when there are only two classes, it is named binary classification)

In the supervised multiclass classification of text, we have a training set of N documents (documents are the text unit considered) with a class: $X_{train} = \{(d_1, c_1), ..., (d_N, c_N)\}$. Our goal is to learn a classifier C that is capable of mapping a new document d to its correct class $y \in Y$, where Y is the set of n classes $Y = \{y_1, y_2, ..., y_n\}$. Thus, the goal would be, at inference time, the model to correctly classify a document $y = C(d)$, where $y \in Y$ and $d \notin X_{train}$.

2.3.1 Class Imabalance Problem

When the number of classes is very large, the classification task becomes more difficult because the model has to learn to differentiate between a greater number of subtle distinctions. Some classes may not be represented adequately in the training data, leading to poor performance. This problem often requires augmenting the amount of labeled data and more complex models.

Numerous strategies have been developed to address the issue of imbalanced datasets, focusing on interventions at either the data level or the algorithm level. Data-level strategies seek to alter the skewed distribution of the classes within the dataset by employing various re-sampling techniques, such as oversampling the less-represented class, undersampling the more dominant class, or combining both to achieve a more balanced dataset. On the other hand, algorithm-level strategies focus on adapting the machine learning algorithms themselves to reduce their predisposition towards the more frequently occurring classes. These adaptations often involve incorporating cost-sensitive learning, which assigns a greater penalty to misclassifying the minority class, and ensemble methods, which combine multiple models to improve prediction performance across imbalanced classes (TEMRAZ; KEANE, 2022).

2.3.2 Evaluation metrics

A confusion matrix is a table used to describe the performance of a classifier on a test set for which the true values are known. It is a good way to clearly see the model's

performance in each class and analyze how the model is predicting wrong labels. For example, some groups of classes can present an ambiguity (the model mostly predicts one class instead of the other). Analyzing these results can give input on the need for more examples to learn how to differentiate classes.

To compute a single aggregate measure for the model that combines the results of each class is common to use Accuracy or F1 measures. Following the equations.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.3}$$

$$recall = \frac{TP}{TP + FN} \tag{2.4}$$

$$precision = \frac{TP}{TP + FP} \tag{2.5}$$

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{2.6}$$

For imbalanced datasets, accuracy is not a good metric alone since the bigger class dominates it. We can consider the macro average of F1 because it reflects the statistics of the smaller classes, and so it is more appropriate when performance in all the classes is equally important. In macro averaging, we compute the performance for each class and then average over classes. In micro averaging, we collect the decisions for all classes in the confusion matrix and compute results from that table (JURAFSKY; MARTIN, 2023). Macro-averaging treats all classes equally, while micro averaging favors bigger classes (SOKOLOVA; LAPALME, 2009).

2.4 Data Augmentation

The capacity of a machine learning model is its ability to fit a wide variety of functions (GOODFELLOW; BENGIO; COURVILLE, 2016). A model with low capacity may underfit the training set, while high-capacity models have a higher potential to overfit by memorizing properties of the training set that damage the generalization over unseen data. An upper bound limits the difference between a model's training and generalization errors. This upper limit increases when the complexity of the model (model capacity) is increased but decreases as we use more training examples (GOODFELLOW; BENGIO; COURVILLE, 2016).

Regularization is a technique used to prevent overfitting by penalizing and con-

straining complex models. Examples are L1 (Lasso), L2 (Ridge), Weight Decay, Dropout, and Batch Normalization, among others. DA is a form of implicit regularization closely related to explicit regularization techniques such as Weight Decay and Dropout (PELLICER; FERREIRA; COSTA, 2023; ZHAO et al., 2019; BOUTHILLIER et al., 2016). According to Goodfellow, Bengio and Courville (2016), the augmentation of a dataset can dramatically reduce the generalization error of a machine learning technique.

Bishop and Nasrabadi (2006) describe the concept of invariance: predictions should be invariant under some transformations of the input. For instance, in speech recognition, minor amounts of non-linear distortion along the timeline while maintaining the sequence of events ought not to alter the understood meaning of the audio signal.

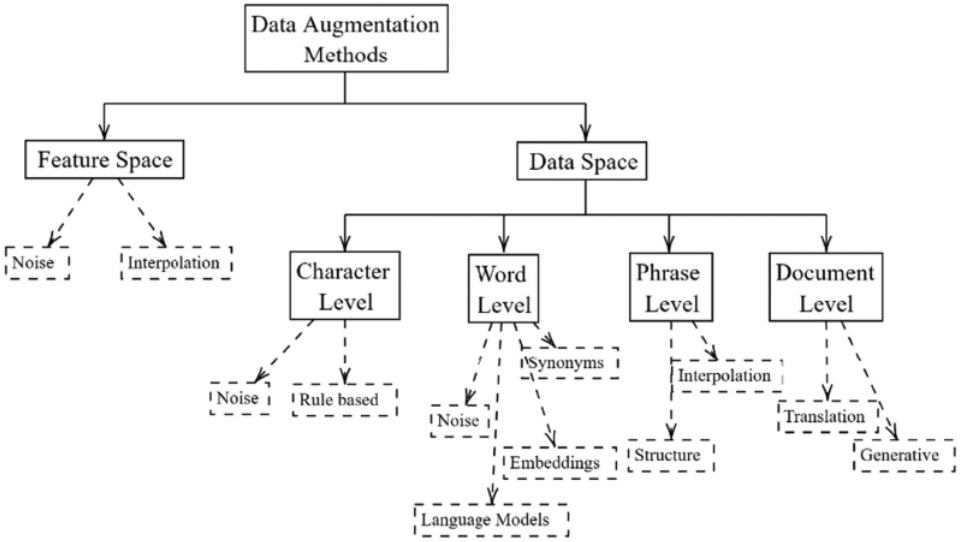
When an adequate number of training examples are provided, adaptive models like neural networks have the capability to learn, at least to a near-accurate degree, the invariance (BISHOP; NASRABADI, 2006). For example, for translation invariance in an image, the training set should include examples of objects at many different positions.

There are several data augmentation methods, and Figure 2.8 presents a taxonomy for DA in text classification. The Feature-based methods refer to transformations of the feature representations of the input (embeddings), such as the Synthetic Minority Oversampling Technique (SMOTE), which is an interpolation method (BAYER; KAUFHOLD; REUTER, 2022).

In Figure 2.8, data space refers to transformations of the input data in its raw form, *i.e.,* into the readable textual form of the data. The transformations are applied over a text unit: character, word, phrase, or document. The dotted squares refer to methods linked to each unit of text, but in some cases, they can be used by more text units. For example, one can apply translation at the phrase level (translation refers to back-translating the text to insert variations on text). Language Models at the word level refer to using a model to replace/insert words. However, language models can generate a word or a sequence based on an input sequence, and this also has to do with the Generative approaches at the Document Level, which can be applied to sentences, too.

The ideal DA strategy should generate augmented data while balancing similarity and diversity in relation to the original set – the goals are model improvement (higher accuracy, for example) and a reduction in overfitting. DA techniques enable the concept of "semantic invariance" by applying label-preserving transformations (HERNÁNDEZ-GARCÍA; KÖNIG, 2018; SHORTEN; KHOSHGOFTAAR; FURHT, 2021). For text classification, we can say that the semantic space to be respected by the DA transforma-

Figure 2.8 – The Transformer architecture



Source: (BAYER; KAUFHOLD; REUTER, 2022)

tions is the one that still ensures the same label assignment to the augmented instance.

DA can potentially aid low-resource languages and can be used to mitigate bias and reduce the class imbalance (FENG et al., 2021) even in resource-rich languages. This thesis focused on experiments transforming the data space. Generative methods received special attention, especially when accomplishing the goal of generating an NLI dataset.

3 RELATED WORK

The related work of this thesis covers three groups of works. Since our investigation path to produce a synthetic NLI dataset studies text generation methods and compares them to DA techniques, we first have Section 3.1 to present some key works about DA techniques for text classification. Secondly, we explored papers that applied text generation techniques to produce synthetic instances for text classification, found in Section 3.2. The last group of works investigated NLI research that produced manual and synthetic datasets, found in Section 3.3.

3.1 DA in Text Classification

We investigate key works on DA for text classification, aiming to facilitate comparing them to text generation methods. The DA methods were selected in an exploratory search, looking for works that are still up to date (recent papers are adapting them or are using them as baselines) and techniques that were easy to implement and seemed straightforward to adapt to Portuguese.

DA refers to introducing new training instances in the process (HARALABOPOULOS et al., 2021), and we focus on some data space approaches mentioned in 2.4. To specify our focus, we roughly divide DA techniques that produce new textual instances into two groups: (i) techniques that augment datasets using labeled data and then use the augmented data in supervised text classification (WEI; ZOU, 2019; AMJAD; SIDOROV; ZHILA, 2020; BODY et al., 2021); and (ii) techniques that also take advantage of unlabeled data using semi-supervised learning or that condition the pre-training of language models (XIE et al., 2020; WU et al., 2019). Our focus in this investigation is on the first group of techniques. The investigated methods are described next.

3.1.1 Back Translation

Back Translation (BT) allows the generation of paraphrases that serve as augmented data (EDUNOV et al., 2018; SENNRICH; HADDOW; BIRCH, 2016). A Machine Translation (MT) system is used to translate an instance from its original language A into language B, and then translate it back to language A. The idea is that the suc-

cessive translations introduce some variation in the text, generating a paraphrase of the original sentence while keeping it in the same class.

Amjad, Sidorov and Zhila (2020) investigated fake news detection in Urdu, a language with scarce resources. They augmented an annotated dataset containing 900 instances with 400 labeled instances from a dataset on the same domain in English and translated them into Urdu. No improvements were observed in the classification task, which was attributed to the poor quality of the automatic translation between these languages.

Beddiar, Jahan and Oussalah (2021) presented a method that fuses BT and Paraphrasing: the original data and the new back-translated data are fed into a paraphrasing model. They used LSTM and CNN classifiers and FastText embeddings to check downstream task results. They ran experiments in five datasets, primarily for binary classification, but one of them had seven classes. They observed improvements in F1-score ranging from 3 to 33% when training classifiers with the augmented data.

Ferreira and Costa (2020) proposed Deep Back-Translation (DeepBT), which adds intermediate translations in N languages between the original text in language A and the final paraphrase also in language A. The N translations lead, at the end, to one transformed instance. Translation happens in different languages, as follows: A $\rightarrow$ L_1 $\rightarrow$ L_2 $\rightarrow$ ... $\rightarrow$ L_N $\rightarrow$ A. Authors evaluated their method on datasets of Microblog messages and News Statements and Headlines. Results showed DeepBT yielded greater performance compared to the traditional BT. However, it was still inferior to EDA when run over small percentages of the dataset.

3.1.2 Contextual Augmentation

Contextual Word Embeddings (CWE) for DA leverages contextual word embeddings from LLM to find similar words and replace them, given the context surrounding the original words. Contextual augmentation was proposed by Kobayashi (2018). The rationale is to replace original words with words predicted by a bi-directional language model. The authors used four binary and two multiclass datasets, with five and six classes. The paper remarks the method is independent of any task-specific knowledge or rules and can be used for classification tasks in various domains. The authors recognized they got a marginal improvement (they got around 1% improvement in accuracy using the augmented datasets) and call for further investigation and comparisons with other methods.

A black box attack was proposed by Garg and Ramakrishnan (2020) for generating adversarial examples using contextual perturbations from a BERT masked language model generating alternatives for the masked tokens. These attacks can serve as an augmentation strategy. Their generated adversarial instances improved grammatical and semantic coherence compared to prior work.

3.1.3 Easy Data Augmentation

Wei and Zou (2019) proposed the Easy Data Augmentation (EDA) strategy. It consists of swapping/deleting words and inserting/replacing words with synonyms, and it does not rely on neural models. EDA relies on four operations, namely, synonym replacement, random insertion of new words that are synonyms, random word swap, and deletion. Each of these operations can be configured as percentage parameters. The authors demonstrated robust results for small datasets. Tests on five datasets augmented with EDA using only 50% of the available training set could achieve the same accuracy as training with all available data. EDA is a simple method widely used as a baseline (YOO et al., 2021; KIM et al., 2022; ANABY-TAVOR et al., 2020).

3.1.4 Fine-tuning LLMs for Paraphrase Generation

Recently, large language models using transformer architectures yielded state-of-the-art performance for several NLP tasks using less annotated data. This is also the case of paraphrase identification and generation (WITTEVEEN; ANDREWS, 2019).

Paraphrase datasets are commonly structured for binary classification, *i.e.,* human annotators assign a label to indicate whether two sentences are paraphrases of each other. The paraphrasing dataset can then be used to fine-tune an LLM that produces a paraphrase given some input sentence. This model, which we refer to LMPG (Language Model for Paraphrase Generation), can be used to augment the training data for several NLP tasks (WANG et al., 2022b).

Fenogenova (2021) compared different language models for paraphrase generation in Russian. They evaluated the quality of paraphrase generation in a two-step procedure: 1) calculating universal metrics (ROUGE-L, BLEU-n) to evaluate the paraphrase quality in a test set; 2) using the models to generate paraphrases as an augmentation

strategy for three different tasks: sentiment classification, textual entailment recognition (it is a sentence pair classification: Entailment/Not Entailment), and question answering dataset for yes/no questions (each example is a triplet of question, passage, and answer). Controversially, the authors did not find improvements when training models with the augmented datasets.

Paraphrase generation was also used as an augmentation technique in (OKUR; SAHAY; NACHMAN, 2022), where a BART model was fine-tuned to generate paraphrases for the Intent Classification task in English, which was the main NLU task from a Multimodal Dialogue System. They improved results on their small-scale task-specific datasets by 4%.

Pegasus is a language model that explores objectives in the pre-training phase tailored for abstractive text summarization (ZHANG et al., 2020). Authors pre-trained large Transformer-based encoder-decoder models on massive text corpora with a new self-supervised objective: important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to extractive summarization. Pegasus achieved state-of-the-art results in the summarization task, and the authors mention its potential for paraphrasing (ZHANG et al., 2020).

3.2 Synthetic Data through Text Generation

According to Li et al. (2023), with the recent advancements in LLMs, there is an increase in the number of research exploring the potential of LLMs for generating synthetic data tailored to specific tasks and then augmenting the training data in low-resourced data settings. In this section, we investigate works that used the text-generation capabilities of LLM to generate synthetic data for text classification tasks.

GPT-2 was used for DA in text classification tasks by (ANABY-TAVOR et al., 2020). The authors fine-tuned the model for the next token prediction using the training instances and labels. The goal was to adapt the model to be able to generate text conditioned to the labels. In real applications with very scarce data, like the ones we experiment with here (in which a class has only eight instances), there may not be sufficient data to successfully incorporate details related to the writing conditioned to that class.

Yoo et al. (2021) used GPT-3.5 for augmentation by taking a few instances from the training data and embedding them in the prompt. Each prompt contained one exam-

ple of each class, and the model was requested to generate one additional instance and classify it in one of the given classes. Five binary classification datasets and one dataset with six categories were used. The authors report improvements in nearly all datasets. However, they used a validation dataset with hundreds of instances. We argue that this resource is not available in most real-world scenarios in which the validation set is usually proportional to the size of the available data.

Ye et al. (2022) argues that synthetic data often suffers from low-quality issues such as low informativeness and redundancy. To address this problem, the authors propose a progressive zero-shot dataset generation framework, PROGEN, with a generator and classifier for the specific task. The generator produces synthetic instances based on a manually designed prompt. Still, they make this process in phases: they generate the first group and train the classifier, and after that, run a feedback function to select the most influential subset according to the loss related to the specific data points in the training of the classifier. The generator model used was GPT-2XL (with 1.5 billion parameters), with a DistillBERT and an LSTM as the classifiers. The results were reported over binary classification datasets of reviews. The model trained with the selected synthetic data could not surpass the supervised model trained with human-annotated data but achieved close results in some datasets. The authors compare results with and without the feedback function, and there is a significant improvement when using the feedback function.

A multilingual classification of news articles into 35 distinct classes from ESG (Environmental, Social, and Governance) in French, Chinese, and English was approached in Glenn et al. (2023). GPT-3.5-turbo was used to produce synthetic instances and augment an imbalanced dataset. The authors selected classes to augment based on the class distribution. The zero-shot strategy was used for non-ambiguous classes and few-shot for ambiguous ones. The results on the classification were mostly higher using the synthetic instances, but not always. Analyzing qualitatively, authors suppose the low variance of the produced instances for some classes was why the dataset augmented with synthetic data was not always the winner. Additionally, many synthetic instances seem easier to classify than the original data for those classes.

Li et al. (2023) remarks that the effectiveness of synthetic data generated by LLM in supporting model training is inconsistent across different classification tasks. The authors used the GPT-3.5-Turbo model to generate instances in a zero-shot and few-shot setting and produced synthetic instances for ten classification datasets. The highest number of classes of these datasets is four. Results indicated that subjectivity, at both the task

and instance level, leads to worse performance of synthetic data, even with high volumes, compared with results with real-world data. Few-shot settings increase data diversity and boost the performance of the resulting models.

3.3 Natural Language Inference

In this Section, we first present works that produced human-generated NLI datasets in English and Portuguese. The goal is to understand the resources available and the process used to create them, described in Section 3.3.1. Finally, in Section 3.3.2, we explored works that produced synthetic instances for NLI.

3.3.1 Human-Generated resources for NLI

We investigated the prominent NLI datasets in English and Portuguese. We describe the macro processes used to generate them, trying to give an idea of the number of persons participating and the role they played.

The Stanford Natural Language Inference (SNLI) (BOWMAN et al., 2015b) dataset is a collection of 570k pairs of sentences labeled for entailment, contradiction, and semantic independence. Premises were collected from Flickr30k, a dataset with image captions (YOUNG et al., 2014a), that contains literal descriptions of scenes. We argue this makes a good source of premises. The authors used Amazon Mechanical Turk to collect the hypotheses, and about 2,500 workers contributed to the task. The premises were presented to the workers, who were asked to write hypotheses for each label (entailment, neutral, and contradiction). Four annotators revised ten percent of the pair labels, achieving agreement in almost 98% of the revised sample.

Bowman et al. (2015a) highlighted that NLI classification is challenging due to indeterminacies of event and entity coreferences, which lead to disagreements in the annotation process about the semantic label, even with human annotators. SNLI authors present some examples of this: Consider the sentence pair *"A boat sank in the Pacific Ocean"* and *"A boat sank in the Atlantic Ocean"*. It could be labeled as a contradiction if one assumes that they refer to the same single event but could also be reasonably labeled as neutral if that assumption is not made. Regarding entity coreference, consider the pairs *"A tourist visited New York"* and *"A tourist visited the city"*. If one considers that the city

is New York (entities are coreferent), the label assigned is entailment, and if not, the label is neutral.

Kalouli et al. (2019) evaluated the annotations and annotators' comments in SNLI and found some confusion about when a pair should be a contradiction or neutral. Annotators considered contradiction pairs in which sentences had nothing to do with each other. For example, the pair *"Two sumo ringers are fighting."* and *"A man is riding a water toy in the water"* was labeled as a contradiction, with the justification "the subjects and activities are completely different". The annotator's error was to confuse the absence of a coreferential relationship with a contradictory one. However, the lack of coreference or connection between two statements should lead to a judgment of neutrality in NLI, not contradiction.

Another important English dataset is the Multi-Genre Natural Language Inference (MultiNLI) (WILLIAMS; NANGIA; BOWMAN, 2018), which contains 433k pairs of sentences. Premises were derived from ten sources representing ten different genres (government, letters, fiction, *etc.*). Generating the hypotheses was similar to SNLI, and four annotators confirmed each label. The authors argued that this diversification captures more of the complexity of modern English.

Portuguese NLI datasets are usually translations. The dataset SICK-BR (REAL et al., 2018) is the manually revised translation of the English dataset Sentences Involving Compositional Knowledge (SICK)(MARELLI et al., 2014). Ten annotators thoroughly revised all translations. An online tool was used in the annotation process and annotators looked at each other's work when translating their sentences. Each annotator could also mark complex sentences that they thought needed further review or where they did not want to review. The authors' main goal was to keep pairs of sentences from English and Portuguese aligned as much as possible, preserving the labels.

ASSIN2 (REAL; FONSECA; OLIVEIRA, 2020) is the Portuguese acronym for Evaluating Semantic Similarity and Textual Entailment and it corresponds to an NLI dataset with two classes: entailment and non-entailment. It used SICK-BR data and, aiming at balanced classes, authors used a semi-automated strategy, taking SICK-BR pairs annotated as entailment and changing some synonyms or removing adverbial or adjectival phrases. They also created new pairs for the entailment class. All generated pairs were annotated by at least four native speakers of Brazilian Portuguese with linguistic training. Only pairs annotated with the same label by most annotators were included in the dataset.

3.3.2 Synthetic Data for NLI

We investigated recent works producing synthetic NLI instances either automatically or semi-automatically. In special, we considered works that use the text-generation capabilities of LLMs.

Text generation was explored by Sadat and Caragea (2022) to create hypotheses for selected premises obtained from other NLI datasets in English. The authors proposed a semi-supervised learning (SSL) framework. First, they fine-tuned BART models with a small set of pairs, conditioning them to produce hypotheses for each class: the premises from the selected pairs are used as the source texts, and their hypotheses as the targets. Ultimately, they got one conditioned BART^C model per class C. With BART^C they generated hypotheses for given premises and assigned the pseudo-labels of each class C. For quality assurance of the generated pairs and labels, they used an interactive process where classifiers are trained with the first selected pairs and predicted on the produced pairs, selecting only instanced with high-confidence predictions and adding them to the labeled dataset. The process is repeated until a defined limit. They selected only instances with high-confidence predictions and added them to the labeled dataset, repeating the process until a defined limit. They compared results on the produced instances with BERT models trained with fully human-annotated datasets, with some data augmentation methods and other SSL methods getting superior performance.

Meng et al. (2022) generated synthetic data for English using class-conditioned texts based on prompts with GPT-XL (1.5B parameters). The resulting instances are then used to fine-tune the bidirectional RoBERTa large (356M parameters) as the classifier. The approach, named SuperGen, includes strategies like quality data selection based on generation probability and regularization techniques for better generalization and stability of the classifier. The generation is done by providing the LLM with prompts that describe the task and the desired outputs. Authors use the term "zero-shot learning" to refer to the fact that the method does not require any human-annotated task-specific data for training the bidirectional LLM. Instead, it relies solely on synthetic data. The results are reported on GLUE, comparing the outcomes using only synthetic data with the conventional training using annotated data. Specifically, for the NLI task, they used the dataset MultiNLI (WILLIAMS; NANGIA; BOWMAN, 2018), and the model trained only with synthetic data achieved around 66.1% of macro-F1, while the baseline using annotated data yielded 81.7%.

Liu et al. (2022) approach the problems of lack of diversity in big NLI datasets as MultiNLI. Authors advocate that a challenge with large crowdsourcing datasets is that human writers are often few compared with the number of produced instances. Additionally, workers (people) use repetitive patterns when preparing examples. It causes models to overfit to such repetitive patterns and fail to generalize to out-of-domain instances. They propose an approach for dataset creation based on worker and AI collaboration, composed of four stages and counts on a starting big dataset. In the first stage, they execute a data map to locate ambiguous examples (counterexamples). Authors advocate counterexamples take more epochs of training to be learned and are crucial for generalization, leading to more robust models. In the second stage, they used in-context learning with GPT3-Curie. Similar examples were sampled from the ambiguous set and prompted to the model. The third stage executes automatic filtering, removing cases where GPT-3 generated an equal premise and hypothesis or copied the example in the few-shot prompt. Using a newly proposed metric, they filtered examples, keeping the most ambiguous. The fourth and last stage adds humans in the loop to review labels and optionally suggests discarding examples or rewriting them. Amazon Mechanical Turk was used in this stage, ensuring two workers per instance. Results of the model trained on the new dataset compared to the original MultiNLI in several English out-of-domain datasets are significantly higher, even having only 25% of the size of MultiNLI.

Akoju et al. (2023) used data augmentation techniques to produce a new synthetic dataset with 1,304 sentence pairs created by modifying 15 examples from the SICK English dataset. They used a variety of modifiers (universal quantifiers, existential quantifiers, negation, *etc.*). Results were evaluated using NLI models trained in other datasets to predict the instances and fine-tune the same models on the new dataset. They did not find significant differences in the results, but analyzing the predictions, they observed that instances modified with adjectives, adverbs, and universal quantifiers performed better than sentences modified with negation and existential quantifiers. Additionally, the authors report that models seem confused when the label is Neutral, and the modifier types are negations.

3.4 Summary and Discussion

Most methods investigated for augmenting text classification datasets are easy to implement, and there are ways to adapt them to Portuguese using translations. They are

important to give the first step towards producing the NLI dataset since they establish a baseline to comprehend the potential for text generation as a DA method.

The works applying text generation as an augmentation technique showed this research area has received increasing interest. Several of the works used GPT models in zero-shot and few-shot settings combined with filtering methods to select the best examples to be part of the prompt or to filter the outputs. The vast majority of works focus only on English. But GPT-3 has been applied to French, a language with the same root as Portuguese. Another interesting point with GPT-3 is the API service, which helps bypass infrastructure limitations, but one must pay for it.

We can conclude from the investigated NLI resources assembled manually in English and Portuguese that they involved a vast manual effort to produce important datasets. Also, the processes of extracting premises from existing sources and then human writing the hypothesis prevails. The NLI manual annotation process is susceptible to indeterminacy issues caused by individual annotators' assumptions about coreferences. This leads to disagreements regarding the neutral class and the other two (entailment and contradiction). For the Portuguese datasets, the major investment was in reviewing and correcting translations and labels since almost all instances came from an English dataset.

When looking at works that produced NLI synthetic instances, we see that Sadat and Caragea (2022), Meng et al. (2022), and Liu et al. (2022) are close to ours since they approached the creation of NLI datasets automatically using text generation. Meng et al. (2022) uses GPT-2XL for English. In this work, we want to go for a model that handles the Portuguese well and allows us to run it in an affordable infrastructure. Our work differs from Sadat and Caragea (2022) because we have not fine-tuned one language model to be conditioned to each class. We took advantage of GPT-4's ability to follow instructions due to its training on reinforcement learning from human feedback (CHRISTIANO et al., 2017), and using an in-context learning strategy (BROWN et al., 2020) designing prompts with qualitative selected examples at inference time. Regarding evaluation, we also differ from them since we manually revised all the generated pairs, which gave us a real perception of the quality of our dataset. Liu et al. (2022) was focused on generating a highly generalized dataset using challenging pairs. To get those pairs, they map the large NLI dataset MultiNLI. In our case, this is still not feasible with the small existing sets we have in Portuguese. But, inspired by their work, we can build a good and challenging list of examples to be used in the prompt to generate instances.

4 SYNTHETIC INSTANCES FOR TEXTUAL CLASSIFICATION

In this chapter, we answer the following research questions:

RQ1. Which augmentation strategies should be selected for text classification to pave the understanding of generative approaches to produce synthetic instances?

RQ2. Besides improving the final classification results, what else is important in DA to ensure high-quality synthetic instances?

RQ3. Is text generation competitive for DA? What are the pros/cons compared to other methods? What insights acquired exploring DA can be investigated in the next step to produce synthetic instances for NLI in Portuguese?

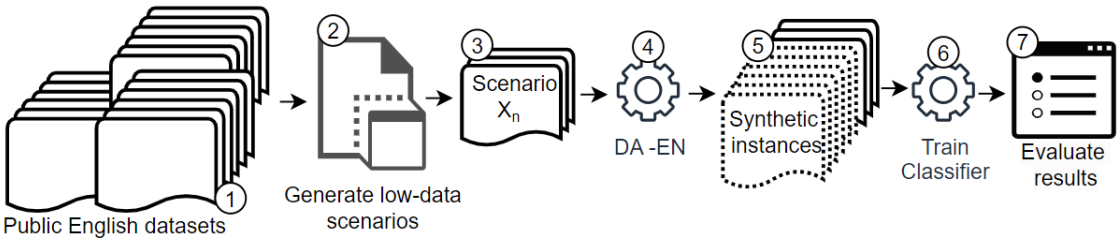
We investigate DA for standard text classification in two languages: $i)$ in English, sentiment polarity and news category attribution are studied in low-data scenarios; and $ii)$ in Portuguese, a use case is presented with a multiclass dataset with tweets labeled into 16 classes related to Smart City concepts and that we submitted to augmentation strategies adapted to Portuguese.

Figure 4.1 presents the two stages of this investigation. The selection of the techniques was based on an exploratory investigation considering traditional methods and also generative options, which we hypothesize could later be used to generate synthetic instances for NLI. Our selection also considered options that seemed promising and easy to adapt to languages other than English, *i.e.,* Portuguese.

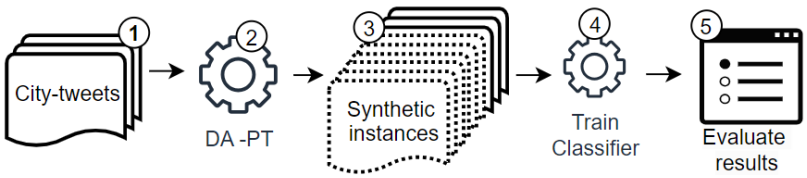
Figure 4.1a describes the high-level steps executed in the first stage. Starting with selecting public English datasets (1) and then the simulation of low-data scenarios sampled from the datasets (2). For each scenario X_n (3), where X is the scenario index generated using seed number n, DA algorithms were run (4), and synthetic instances were produced (5). A classification model was trained using an augmented version of the scenario generated by each DA method (6), and results were evaluated by comparing them with a baseline run with and without augmentation and with the results obtained when training only on the synthetic instances (7).

Figure 4.1b depicts the second stage of the investigation of DA for text classification, where some of the studied methods in English were adapted to Portuguese and applied to a real use case. A dataset called City-tweets (1) was assembled containing 16 classes that represent the Smart City dimensions each tweet refers to (*e.g., transportation, energy, water, etc.*). The studied DA methods were adapted to Portuguese and applied (2)

Figure 4.1 – DA for text classification



(a) DA in English datasets which were organized in several low-data scenarios



(b) DA in a Portuguese dataset

Source: the author

to generate synthetic instances for each small class in City-tweets (3). For each method, a classifier was trained (4), and results were evaluated (5). Besides the quantitative results, we also performed a qualitative evaluation. Results were analyzed, looking for paths to adapt the text generation scripts used in DA to our final goal: produce synthetic instances for NLI.

We use GPT models in both experiments to augment the dataset using text generation capabilities, looking for paths to develop the synthetic NLI dataset in Portuguese.

4.1 English Datasets in Low-data Regimes

Since we are interested in scenarios with very scarce data, our approach was to simulate low-data scenarios using well-known datasets for text classification. As expected, the language in which most available datasets are written is English. We aimed to understand how classification results behave with different class sizes while keeping the same number of instances across classes.

To put the task more formally, we take a classification algorithm A trained on a dataset $D_{train}\{(x_i, y_i)\}_{i=1}^{n}$ with n instances, where each instance x_i labeled with its class y_i to generate a classifier $C = A(D_{train})$. C is then evaluated on a test dataset $D_{test}\{(x_i^t, y_i^t)\}_{i=1}^{z}$ with z labeled instances, where $D_{train} \cap D_{test} = \emptyset$. We evaluate the success of the classifier C over D_{test} with the function $T(C(x_i^t), y_i^t)_{i=1}^{z}$ which compares

the classifier predictions $C(x^t)_i$ against the respective ground truth label y_i^t.

In this work, we are not focusing on the classification algorithm but on the DA method M, which produces an augmented artificial dataset $D_{synthetic}$, *i.e.,* $D_{synthetic} = M(D_{train})$. The goal is to get an improved classifier $C^{'}$ when training algorithm A in the augmented version of the dataset $C^{'} = A(D_{train} \cup D_{synthetic})$, which contains the original and the new synthetic instances. In this way, we can evaluate the impact of M in $C^{'}$ compared to C.

In the next subsections, we describe the datasets, the DA methods, our experimental setup, and the results.

4.1.1 Datasets

The following datasets were used in our experiments. Their statistics are in Table 4.1.

- **AG-news** (GULLI, 2005). A dataset of news articles collected from various sources. We adopted the annotation by (ZHANG; ZHAO; LECUN, 2015), keeping only the four largest classes.
- Stanford Sentiment Treebank (**SST-2**) (SOCHER et al., 2013). A sentiment analysis dataset with sentences from movie reviews. We used the Binary Classification classes available here[1].
- **TweetEval** for Sentiment (ROSENTHAL; FARRA; NAKOV, 2017). TweetEval has annotations for seven classification tasks. We used the sentiment labels with three classes.

4.1.2 Dealing with Multiple Sentences

For EDA, LMPG, and CWE (described previously in Sections 3.1.3, 3.1.4, and 3.1.2 respectively), we augmented over single sentences. However, in some cases, the instances from the datasets used in our experiments had more than one sentence. The proportion varies across datasets and can be seen in Table 4.1. In TweetEval, almost half the instances had more than one sentence. In AG-news, multiple sentences were found in about

[1] <https://github.com/YJiangcm/SST-2-sentiment-analysis/tree/master/data>

Table 4.1 – Statistics for the English Datasets

Dataset	Classes	Train	Test	% one-sentence
AG-news	World	30,000	1,900	76.16%
	Sports	30,000	1,900	
	Business	30,000	1,900	
	Sci/Tech	30,000	1,900	
SST-2	Positive	4,054	909	99.40%
	Negative	3,738	912	
TweetEval	Positive	17,849	819	50.62%
	Neutral	20,673	869	
	Negative	7,093	312	

Note: number of instances in the train/test split by class, and the percentage of the total instances that contain only one sentence.

23% of the instances. In SST-2, however, nearly all instances had single sentences.

For all DA methods, except Back Translation and Text generation, the following procedure was used. First, we split each document into sentences and ran the augmentation for each sentence in isolation. Then, the augmented instances are created by concatenating the augmented versions of their composed sentences. For example, let D_0 be an instance to be augmented containing two sentences: $D_0 = [a_0, b_0]$. Sentence a_0 generated two distinct augmented sentences $augS_{a_0} = [a_1, a_2]$ while sentence b_0 yielded only one $augS_{b_0} = [b_1]$. To generate the final augmented instances D_{aug} for the original instance D_0 we added a second instance into the augmentation list using the original sentence $augS_{b_0} = [b_1, b_0]$. The final set of augmented instances is $D_{aug} = [d_1, d_2]$, where $d_1 = [a_1, b_1]$ and $d_2 = [a_2, b_0]$.

4.1.3 Experimental Procedure

In this section, we first describe how we simulated the low-data scenarios. Then, in Section 4.1.3.1, we detail the DA methods describing models and settings used to generate the synthetic instances. Finally, in Section 4.1.3.2, we explain how the classifiers were trained and evaluated.

For each dataset, we randomly sampled 200 instances from each class of the training set. The test set is kept untouched to avoid data leakage. From those 200 instances, we generated smaller sets, removing 25 instances per step until there were at least ten instances by class. At the end of the process, we obtained nine different sets containing the following numbers of instances by class 10, 25, 50, 75, 100, 125, 150, 175, and 200. We

repeated this process five times using different seeds, resulting in 45 low-data scenarios.

4.1.3.1 Configuration of the DA Methods

We applied different DA techniques in the low-data simulation to understand their potential. The rationale of the DA techniques is given in Section 3. Here, we describe the configurations used in each of them. Since the DA methods can generate repeated augmented versions of the same sentence, duplicate removal is required. Sentences were compared using their tokens. In the comparison, we used NLTK to generate word tokens and discarded stopwords, symbols, punctuation, and numbers.

- **Back Translation (BT)**: We used the free service of Google Translate through the DeepTranslator[2] library. We ran four translation operations generating four potential augmented instances: we back-translated them using Portuguese, German, Russian, and Arabic. The rationale for the choice of languages was to get diversity. Thus, we picked languages from different families: English and German, from the same root, and Russian and Arabic, both from different roots. The entire instance was translated (*i.e.,* no sentence splitting was performed). Since different target languages may yield the same back-translated text, removing duplicate augmented versions was necessary.
- **Easy Data Augmentation (EDA)**: We used the default configuration that modifies 10% of the tokens for each operation (replacement, insertion, word swap, and deletion). We request to generate ten potential augmented instances for each training instance. The algorithm balances the generation of synthetic instances among the four operations, so each synthetic instance was generated by applying only one operation. We used the original EDA algorithm[3].
- **Contextual Word Embeddings (CWE)**: We used the library NLPaug[4] with the *substitute* action at word level. We set the percentage of words to be augmented to 30% for AG-news and 10% for the other datasets. The difference is due to the texts in AG-news being twice as long as in the other datasets, affecting the results. We also employed a list of stopwords to be skipped from the augment operation. We used embeddings from bert-base-uncased[5] (DEVLIN et al., 2019). For each

[2]<https://github.com/nidhaloff/deep-translator>
[3]<https://github.com/jasonwei20/eda_nlp>
[4]<https://github.com/makcedward/nlpaug>
[5]<https://huggingface.co/bert-base-uncased>

original instance, we generate ten potential synthetic versions.

- **Language Model-based Paraphrase Generation (LMPG)**: We used a Pegasus (ZHANG et al., 2020) model fine-tuned on paraphrase identification datasets such as (ZHANG; BALDRIDGE; HE, 2019). We used `pegasus_paraphrase`[6] to generate up to ten potential augmented instances.
- **Text generation with GPT2 (GPT2)**. We assume that in many real-life application domains, such as sentiment analysis (SST-2 and TweetEval datasets) and news classification (AG-news), unlabeled data is abundant. Thus, we fine-tuned `gpt2-small` in the task of Causal Language Modeling. The complete training data available (see Table 4.1) was used for each dataset. The fine-tuning process was executed in an affordable hardware configuration. The outcome is a new model adapted to the text style used in the specific dataset. This process was done for the three English datasets, resulting in three models used to generate new instances in the low-data scenarios. We generated the synthetic instances using *prefix prompting*, which continues a string prefix (LIU et al., 2023). Our prompt contained the original instance followed by the phrase [",*in other words,*"] to generate a new piece of text. As a decoding strategy, we used top-p sampling (HOLTZMAN et al., 2019), with $p = 0.5$, and top-k sampling (FAN; LEWIS; DAUPHIN, 2018), with $k = 40$. We used GPT-2 small, and when running preliminary experiments, despite the fine-tuning we did for each dataset, we observed that the most probable sentences did not preserve the label and varied too much in semantics (sometimes writing something very distant from the context used as input). Thus, we decided to keep the decoding strategy and generate many more instances filtering for the ten most similar to the original instance. When analyzing the results, we may confirm the average similarity is still low for GPT-2, confirming that it still generates good diversity.
  We generated 150 synthetic instances for each original instance and calculated the cosine between their embeddings using the sentence-transformers (SBERT) model `all-mpnet-base-v2`[7] (REIMERS; GUREVYCH, 2019). We chose the ten synthetic instances with the highest similarity in relation to their corresponding original instance. Duplicates were removed as described at the beginning of this section.
- **Text generation with GPT-3.5**. We used GPT-3.5 models for the English datasets

[6]<https://huggingface.co/tuner007/pegasus_paraphrase>
[7]<https://www.sbert.net/docs/pretrained_models.html#model-overview>

only for the smallest scenarios (with 10 and 25 instances per class) due to budget restrictions. Unlike all other approaches used here, GPT-3.5 is not free. We made an estimate of the cost for running the complete set of experimental runs, which involve five seeds, nine sets of original instances per class size, and three datasets, adding up to 135 experimental runs. GPT-3.5 prices are calculated per 1k tokens sent in the prompt plus the number of tokens received in the responses. Our experimental runs would add up to 693 million tokens. At the time of writing this thesis, the cost for 1k tokens was USD 0.02. Thus, performing the complete experiment would amount to USD 13,860. Due to budget constraints, we were only able to apply GPT-3.5 to the settings with 10 and 25 original instances per class.

We used the *IntructGPT* model called `text-davinci-003`, applying the same augmentation approach and decoding settings used in (YOO et al., 2021). We did not use their training approach nor the full validation set, and we applied only the augmentation phase for the sake of comparability with the other models.

4.1.3.2 Training and Evaluation

The text classifiers were implemented by fine-tuning BERT models for each low-data dataset, adding a linear layer on top of the pooled output. Our choice was motivated by the fact that this is the state-of-the-art in text classification (MINAEE et al., 2021). Fine-tuning used the pre-trained neural network weights from `bert-base-uncased` as initialization. The model has the same architecture as the original BERT-base (DEVLIN et al., 2019): 12 layers, 768 hidden dimensions, 12 attention heads, and 110M parameters. The pre-trained objectives were also identical to the original BERT, *i.e.,* models were trained on Masked Language Modeling and Next Sentence Prediction tasks. In the tokenization process, we used `max_tokens` as the largest number of tokens per instance. For AG-news, `max_tokens` was set to 256 since only 0.4% of the instances were over that limit.

The optimizer used was AdamW (LOSHCHILOV; HUTTER, 2019), which is the same as Adam (KINGMA; BA, 2015) with a weight decay regularization. We set the learning rate to 4e-5 and the dropout to 0.1. The model was trained for ten epochs using a batch size of 32 and selecting the best model at the end (lowest validation loss). We stopped training early when no improvements were achieved after three steps. We did not vary the hyperparameters since our goal was to focus on the impact of DA.

The training sets were split into training and validation (80-20). The classifiers

were evaluated over the original test sets unseen during training. We measured results through Macro-F1 and Micro-F1. The augmented data can introduce some small variations in class balance. Therefore, we report results mainly using Macro-F1 since it allows us to get a sense of the effectiveness on smaller classes (MANNING; RAGHAVAN; SCHüTZE, 2009).

We analyzed the DA methods considering three experimental settings. In the first, we use the union of original and synthetic instances ($D_{train} \cup D_{synthetic}$) to fine-tune BERT. This setting is labeled with `O+S`. In the second setting, we consider only the synthetic instances ($D_{synthetic}$) created by the DA methods. This setting is labeled with `S`. There are also baseline runs where only the original instances are used (`O`).

4.1.4 Results

In this section, we present the results of the English experiments. First, we analyze the augmentation statistics in Section 4.1.4.1. Secondly, in Section 4.1.4.2, we present the macro-F1 results of each scenario and DA method trained with original and augmented instances and only with augmented instances. In Section 4.1.4.3, the results of small scale experiments generating text with GPT-3.5 can be found. We ran it only for the scenarios with 10 and 25 instances per class. Finally, in Section 4.1.4.4, we analyze the processing times.

4.1.4.1 Augmentation Statistics

We calculated statistics for the DA methods, including the average number of tokens, text length, and the cosine similarities in relation to the original instances using sentence embeddings from Sentence-BERT[8]. These statistics are shown in Table 4.2. The difference in the number of instances for each DA method is a consequence of duplicate removal. We can see that LMPG yielded the most duplicated instances (almost 40% in SST2). On the other hand, CWE was able to generate the highest number of distinct augmented instances. The number of average tokens and text length is almost the same across most DA methods. The exceptions were LMPG and GPT2, which got fewer tokens. This behavior for LMPG is expected since the model was trained with an objective similar to summarization. For GPT2, this was caused by the input context combined with the

[8]<https://www.sbert.net/>

Table 4.2 – Augmentation statistics for the low-data scenarios considering 200 original instances.

	Instances	Sim	Toks	Len
	AG-news			
O	800	–	43	234
BT	3,144	0.941	43	231
EDA	6,431	0.924	40	233
CWE	8,000	0.898	46	238
LMPG	5,403	0.824	30	161
GPT2	7,990	0.640	20	98
	SST2			
O	400	–	19	104
BT	1,411	0.881	20	105
EDA	2,682	0.869	20	108
CWE	3,932	0.781	17	103
LMPG	2,418	0.839	17	87
GPT2	4,000	0.494	15	68
	TweetEval			
O	600	–	24	108
BT	2,129	0.915	24	110
EDA	4,173	0.833	22	108
CWE	5,968	0.804	25	109
LMPG	3,868	0.783	20	90
GPT2	5,935	0.461	18	71

Note: The “O" row refers to the original instances. The remaining rows show the number of synthetic instances produced by each DA method, averaged across the five seeds. “Sim” is the average cosine similarity between synthetic and original instances, “Toks” is the average number of tokens per instance, and “Len” is the average number of characters per instance.

size limit we requested as a response (we defined max tokens as the size of the original instance times two, expecting to generate a synthetic instance with a similar size). We also calculated the average similarity of the augmented instances and the original ones. BT and EDA generate less variability since they have high average similarity values. GPT-2 has a very low value for similarity, indicating that even filtering over 150 instances to keep only the ten most similar to the original instance did not affect the diversity.

The number of distinct synthetic instances produced by the DA methods plus the original instances are shown in Figure 4.2. We can see a similar pattern across all datasets – CWE and GPT2 generated the largest number of distinct instances, while BT generated the fewest. This is expected since BT generates only four augmented instances (since we used four languages), and back translation often generates similar instances that are removed by our duplicate removal step. No correlation exists between the number of distinct instances produced and classification quality. The capability of generating more instances has to do with diversity, which can result in semantic drift and negatively impact

classification.

4.1.4.2 Impact of DA methods on classification quality

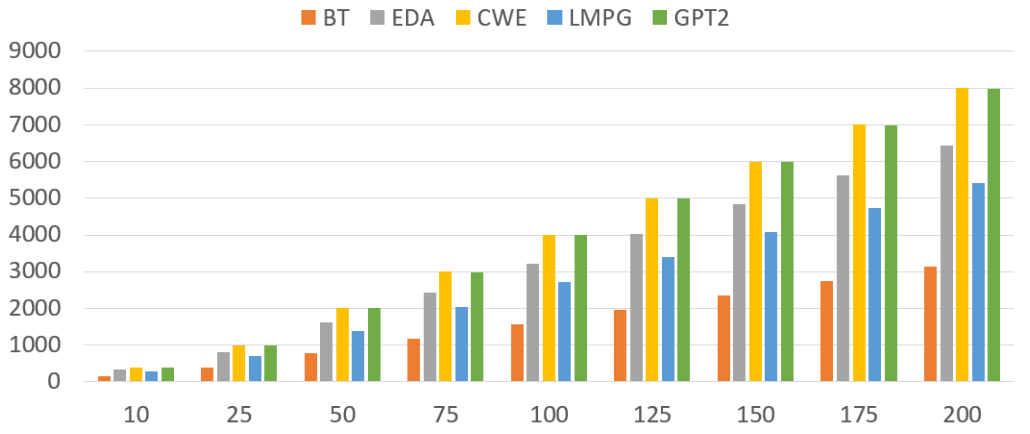
The performances of all methods along the different class sizes in the different experimental settings (O, S, and O+S) are shown in Figure 4.3 and in Table 4.3. The results represent Macro-F1 scores averaged across all five seeds. The number of original instances varies from ten to 200 per class. These original instances were used by each DA method to produce synthetic instances. For example, let us consider the last data point for the BT method in Figure 4.3a, which corresponds to 200 original instances per class. The classification algorithm was trained on approximately 3800 instances, where 800 are original instances (200 per class $\times$ four classes) and 3K instances were generated by BT. Then, for the results in Figure 4.3b, the classifier was trained only with the 3K instances generated by BT.

Figure 4.3, shows that all DA methods yielded important improvements when there were up to 100 instances per class. In SST-2, this can be observed in up to 125 instances and, in TweetEval, across all class sizes. In AG-news, there were no significant performance improvements with DA when class sizes were larger than 100 instances.

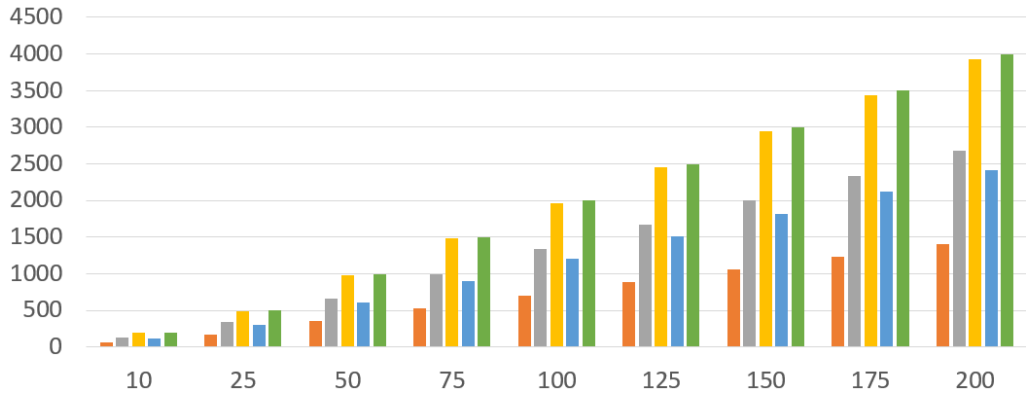
Considering the performance of the different datasets along experiments only with the synthetic data (S), we see that the quality of DA was lower in TweetEval. In that dataset, in 18 of the 45 experimental runs (combinations of nine dataset sizes and five DA methods), the classifier trained on O+S yielded results that were five percent superior (proportionally) compared to the results achieved when training only on synthetic instances (S). In these 18 runs, the biggest differences are concentrated at 50 or fewer instances per class, and GPT2 presents the worse results among the classifiers trained only with synthetic instances in TweetEval. In SST2, only four classifiers trained just with synthetic instances yielded less than 5% of the results obtained on O+S; it happened for EDA, CWE, and LMPG methods. AG-news presents the most stable results. We hypothesize that more formal and long texts bring that stability, while shorter texts, as found in SST2, tend to have lower results when using only synthetic data in a few experiments. Besides the shortness of messages, in TweetEval, there are additional challenges related to informality, irony, slang use, etc., which may cause some low results we saw in some only-synthetic experiments.

The results achieved by the different DA methods were very similar. To have a sense of which method is best, we computed how many times each method had the best

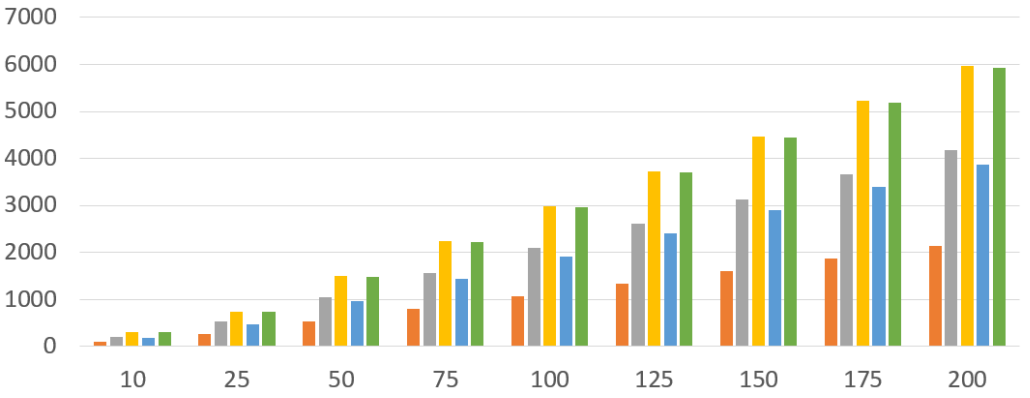
Figure 4.2 – Number of instances used for training the classifier with data generated by the different DA methods



(a) AG-news



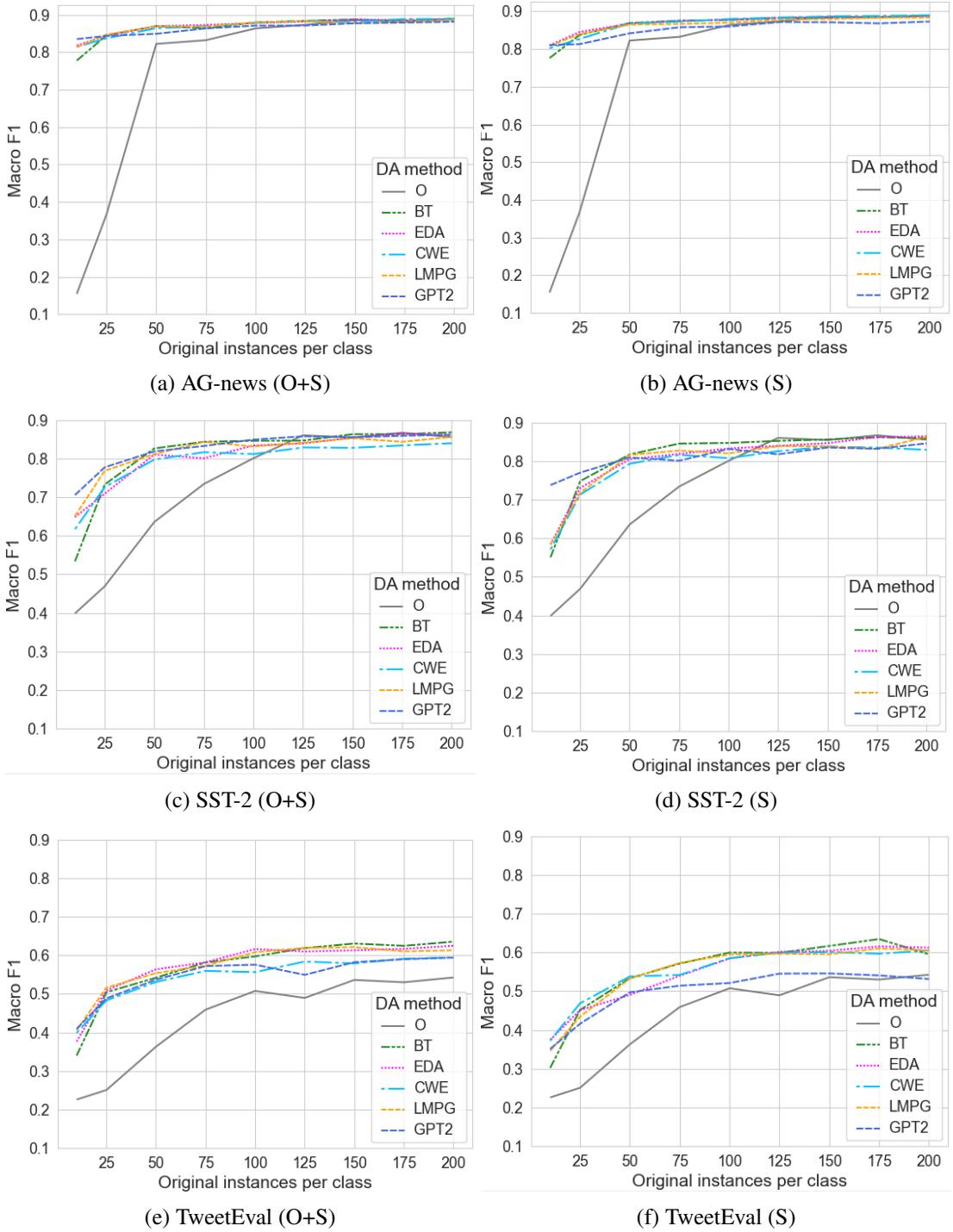
(b) SST-2



(c) TweetEval

Note: The y-axis corresponds to the number of instances used in the training considering synthetic and original ones (S+O). The X-axis refers to the number of instances per class in the original dataset.

Figure 4.3 – Macro-F1 scores for the different low-data scenarios



(a) AG-news (O+S)

(b) AG-news (S)

(c) SST-2 (O+S)

(d) SST-2 (S)

(e) TweetEval (O+S)

(f) TweetEval (S)

Note: We considered two experimental settings: using a combination of the original and synthetic instances (S+O), using only the synthetic instances (S), and the original run with no augmentation (O). The x-axis contains the number of original instances used by the different DA methods

Table 4.3 – Macro-F1 results for the DA methods.

(a) AG-news

Original Instances	O	BT		EDA		CWE		LMPG		GPT2	
		O+S	S	O+S	S	O+S	S	O+S	S	O+S	S
10	0.15	0.78	0.78	0.82	0.81	0.81	0.80	0.81	0.81	0.84	0.81
25	0.37	0.85	0.84	0.84	0.84	0.84	0.83	0.85	0.84	0.84	0.81
50	0.82	0.87	0.87	0.87	0.87	0.86	0.87	0.87	0.86	0.85	0.84
75	0.83	0.86	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.86	0.86
100	0.86	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.87	0.87	0.86
125	0.87	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.87	0.87
150	0.89	0.89	0.88	0.89	0.88	0.88	0.89	0.88	0.88	0.88	0.87
175	0.88	0.88	0.88	0.89	0.88	0.89	0.89	0.88	0.88	0.88	0.87
200	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.88	0.88	0.87

(b) SST2

Original Instances	O	BT		EDA		CWE		LMPG		GPT2	
		O+S	S	O+S	S	O+S	S	O+S	S	O+S	S
10	0.40	0.53	0.55	0.65	0.58	0.62	0.57	0.65	0.58	0.71	0.74
25	0.47	0.73	0.75	0.71	0.73	0.73	0.71	0.77	0.72	0.78	0.77
50	0.64	0.83	0.82	0.81	0.80	0.80	0.79	0.81	0.82	0.82	0.81
75	0.73	0.84	0.85	0.80	0.82	0.82	0.82	0.84	0.83	0.83	0.80
100	0.80	0.85	0.85	0.83	0.83	0.81	0.81	0.83	0.82	0.85	0.83
125	0.86	0.85	0.85	0.84	0.84	0.83	0.83	0.84	0.84	0.86	0.82
150	0.85	0.86	0.86	0.85	0.85	0.83	0.84	0.85	0.84	0.85	0.84
175	0.87	0.86	0.86	0.86	0.86	0.83	0.83	0.84	0.83	0.86	0.83
200	0.86	0.87	0.86	0.86	0.86	0.84	0.83	0.86	0.86	0.86	0.85

(c) TweetEval

Original Instances	O	BT		EDA		CWE		LMPG		GPT2	
		O+S	S	O+S	S	O+S	S	O+S	S	O+S	S
10	0.23	0.34	0.30	0.38	0.37	0.40	0.37	0.41	0.35	0.41	0.35
25	0.25	0.50	0.45	0.51	0.45	0.48	0.47	0.52	0.43	0.49	0.42
50	0.36	0.54	0.53	0.56	0.49	0.53	0.54	0.55	0.53	0.54	0.50
75	0.46	0.58	0.57	0.58	0.54	0.56	0.54	0.57	0.57	0.57	0.51
100	0.51	0.60	0.60	0.62	0.58	0.56	0.59	0.61	0.59	0.58	0.52
125	0.49	0.62	0.60	0.61	0.60	0.58	0.60	0.62	0.60	0.55	0.55
150	0.54	0.63	0.62	0.61	0.60	0.58	0.60	0.62	0.60	0.58	0.55
175	0.53	0.62	0.63	0.62	0.62	0.59	0.60	0.61	0.61	0.59	0.54
200	0.54	0.64	0.60	0.62	0.61	0.59	0.61	0.61	0.61	0.59	0.53

Note: Original instances (O), Synthetically Created Instances (S), or combination of both (S+O)

Table 4.4 – DA method with the highest score for each class size considering O+S.

Class size	AG-news	SST2	TweetEval
10	GPT2	GPT2	GPT2
25	LMPG	GPT2	LMPG
50	BT/EDA	BT	EDA
75	EDA	BT/LMPG	EDA
100	BT	GPT2	EDA
125	BT	O	BT/LMPG
150	EDA	BT	BT
175	CWE	O	BT
200	BT	BT	BT

score across the different datasets and class sizes. BT has the most wins (12 out of 27), was followed by EDA, GPT2, and LMPG. In Table 4.4, we highlight the method that achieved the best results in each scenario. Although this was not an exhaustive comparison, it shows that BT wins in most cases. However, it did not win in the smallest scenarios, whereas generative approaches did better.

Linking the winning methods and the similarity average in Table 4.2, we cannot affirm that a method that produces the most similar instances in relation to the original ones is the best. However, we can see that BT and EDA have high similarity values and perform best in most scenarios. On the other hand, we see GPT2, which has a very low similarity average but yielded the best results with very small class sizes. Besides, GPT-2 won in more scenarios than LMPG, which has a similarity value of around 60% higher than GPT-2.

4.1.4.3 Small scale experiment with state-of-the-art generative methods

Finally, to gain some insight into the performance of state-of-the-art text generation methods, we performed a small experiment using GPT-3.5. We used it to create synthetic instances for all datasets but considered only 10 and 25 original instances per class.

The results are in Table 4.5. The scores for the S+O and the S settings are very close. In TweetEval, GPT-3.5 had noticeably superior results compared to GPT2. The last two columns show the results of the winning method in Table 4.3. We observe large improvements of GPT-3.5 in relation to other methods in SST2 and TweetEval. In AG-news, however, GPT-3.5 did not outperform the best method, we hypothesize this is related to the length of the texts since AG-news has twice the number of tokens and number of characters than the other datasets, as can be checked in Table 4.2. The superior results

Table 4.5 – Macro-F1 and the number of instances (sum for all classes) generated by GPT-3.5 applied on 10 and 25 original instances per class.

		O+S		**S**		**Winning Method**	
dataset	Original instances per class	Total Instances	F1	Total Instances	F1	Total Instances	F1
AG-news	10	433	0.819	393	0.799	GPT2	0.835
	25	1067	0.840	967	0.820	LMPG	0.847
SST2	10	220	0.838	200	0.832	GPT2	0.705
	25	550	0.866	500	0.858	GPT2	0.777
TweetEval	10	288	0.491	258	0.468	GPT2	0.409
	25	708	0.609	633	0.593	LMPG	0.530

Note: The last two columns show the results for the winning method in Table 4.3

achieved by GPT-3.5 in SST-2 and TweetEval motivated us to use it in our use case presented in the next section.

4.1.4.4 Processing Times

Regarding the cost in terms of processing time, BT is the most expensive method as it requires running the translation step twice for each language. For GPT2, besides the time to generate the synthetic instances (which is similar to the time taken by LMPG and CWE), it requires the similarity calculation of the synthetic instances and the original one, which is used for filtering the best instances. Therefore, GPT2 is the second most expensive, taking nearly 50% of the time required by BT. LMPG and CWE took less than 13% of the time taken by BT. EDA is extremely fast, with augmentation running in a matter of seconds.

4.2 Portuguese Dataset - a Use Case in Smart Cities

The worldwide omnipresence of online social media, such as Twitter (recently renamed as "X", but we refer here as Twitter), has been motivating recent research on the socioeconomic status of a population, demographic information, and social science (ZHAO et al., 2022). For instance, the relationship between the city dynamics on Twitter and the land was analyzed by García-Palomares et al. (2018). According to Herdağdelen (2013), social media is an important source for linguistic and sociological research. On

Twitter, hundreds of millions of messages about people's experiences are shared on a daily basis. Tweets represent a challenge for NLP tasks since they are usually written with ungrammatical sentences with a lot of emoticons, abbreviations, specific terminology, slang, *etc*. (PLA; HURTADO, 2018).

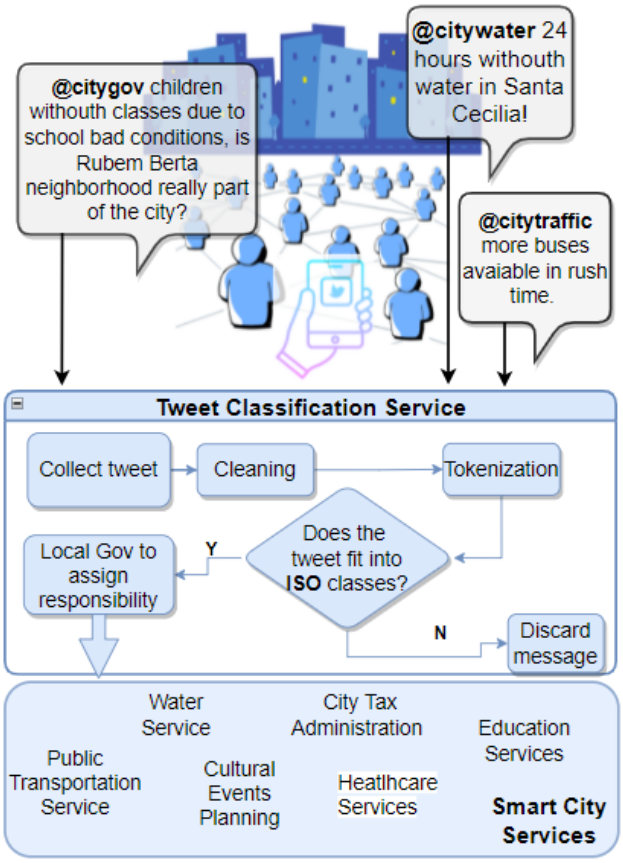
The concept of "Smart cities" is not just a buzzword. "Sustainability" is one of its building blocks and represents a fundamental paradigm in the 21st century to tackle issues such as climate crisis and global warming (PURI; VARDE; MELO, 2022). Smart City models were developed to help city stakeholders agree on what matters when evaluating whether the city can be considered a Smart City. These models recommend evaluating the urban environment considering a set of dimensions (classes). ISO 37120 (ISO, 2014) is one of these models. It recommends a set of indicators to follow up in each city dimension to promote actions that improve that dimension and speed up the city's journey into becoming a Smart City.

In this section, we present a use case covering the training of a classifier for tweets in Portuguese, able to identify the city dimension each tweet refers to. We use the city dimensions of ISO 37120 as our classes: Economy, Education, Energy, Environment, Finance, Fire and Emergency response, Health, Recreation, Safety, Shelter, Solid Waste, Telecommunication, Transportation, Wastewater, and Water. Figure 4.4 presents the motivation of the framework we adopted – monitoring citizens' tweets that mention important Twitter accounts can help improve city services and the urban environment. Once a tweet is classified into an ISO class, the local government would be able to assign responsibilities for taking action in response to the tweet. Citizens do not always communicate using formal channels, but their tweets (either complaining, suggesting, or congratulating) frequently mention Twitter accounts responsible for city services.

4.2.1 The City-tweets Dataset

The City-tweets dataset contains tweets about Porto Alegre, a Brazilian city. Each tweet is labeled with its corresponding ISO 37120 (ISO, 2014) class. Most Twitter users do not release their location, so instead of collecting all tweets from users identifying their location as Porto Alegre, we selected a set of accounts with which the population most interacted about city problems. These accounts include water services, transportation companies, local government official accounts, *etc.*. The accounts encompass public and private companies, newspapers, and civil organizations. We collected tweets that mention

Figure 4.4 – Classification framework for Smart City tweets.



Source: The author

these accounts and added them to the original dataset used by Bencke, Cechinel and Munoz (2020). The goal was to increase the number of instances for some very under-represented classes.

Our dataset contains 1993 tweets labeled with 15 ISO dimensions plus a "*None*" class that applies to tweets that are not related to any of the 15 ISO classes. We removed all mentions to user accounts. We tackle here a classification problem involving more than two classes with each document belonging to only one class, *i.e.,* multi-class single-label classification (MANNING; RAGHAVAN; SCHüTZE, 2009).

Train and test sets are publicly available with 1393 and 600 instances, respectively. The goal is to allow reproducibility and to foster future research in Portuguese. The dataset is highly imbalanced, with most classes having fewer than 60 instances. The number of instances per class in the training set can be found in the column "O" in Table 4.7.

4.2.2 Experimental Procedure

Some DA settings for English described in Section 4.1.3 had to be adapted to work with Portuguese, as follows. The adapted DA techniques were applied only to classes with fewer than 60 instances, representing 12 out of the 16 classes in City-tweets.

- EDA was modified using two approaches, namely **EDA-pt** and **EDA-BT**. **EDA-pt** is the adaptation of the algorithm to handle particularities of the Portuguese language (Portuguese Wordnet[9], stopwords, diacritics). **EDA-BT** is the combination of the original EDA with Back-translation as follows a) translate Portuguese tweets into English, b) run EDA over the English tweets using the original algorithm with the English Wordnet and stopwords, and c) translate the augmented instances from English back to Portuguese.
- **CWE** used the weights from bert-base-portuguese-cased[10] (SOUZA; NOGUEIRA; LOTUFO, 2020).
- LMPG required a translation step before paraphrase generation. We call this method Language Model for Paraphrase Generation with Back Translation (**LMPG-BT**). It consists of the following steps: a) translating the Portuguese instances to English; b) generating paraphrases of the English instances resulting from a; c) translating

[9] <https://wn.readthedocs.io/en/latest/index.html>
[10] <https://huggingface.co/neuralmind/bert-base-portuguese-cased>

back to Portuguese the English paraphrases from b.

- For text generation, since we do not have enough data to fine-tune GPT2 (which was originally trained in English), we experimented with **GPT-3.5**. More specifically, we used `text-davinci-003`, since preliminary evaluations showed it is able to handle Portuguese very well.

  In the English datasets, as described in Section 4.1, we used the GPT-3.5 model following the augmentation process described in (YOO et al., 2021). The authors propose building prompts using $k = 2$ instances from different classes. However, their experiments mainly were over binary classification datasets and one multiclass dataset with only six classes. In our use case, we have 12 classes that need augmentation. We conducted preliminary tests to choose the best k for our setting and noticed that the model generated a very uneven number of instances across classes. Since this imbalance does not serve our purpose, we generated synthetic instances using the following prompt template per class:

  `Prompt per class = class description + instruction + examples`

  Below is an example for the class "Environment":

  - `class description`: "*Nas redes sociais, o tema Meio Ambiente abrange discussões sobre áreas verdes das cidades, mata nativa, desmatamento, sustentabilidade, uso do plástico, arborização da cidade, plantio e podas de árvores, poluição do ar, poluição sonora.*"
  - `instruction`:"*Gere tweets sobre Meio Ambiente:*"
  - `examples`: "Tweet: example1 <CR> Tweet: example2..."

  The instruction is the same for all classes, and it requests the model to generate and return all choices. Choices are possible predictions the model generates given a specific input prompt. When those experiments were run, the maximum number of choices allowed for all paid models in OpenAI was 128. For each class, we sent only one request. This was possible because the number of tokens in the prompt plus each choice response was always below the limit of the model (*i.e.,* 4,096 tokens for GPT-3.5 and GPT-3.5T, and 32k tokens for GPT-4). The largest class is *Health* with 35 instances, and the prompt for that class has 1,192 GPT tokens and 5,299 characters. Regarding the decoding settings, we used $temperature = 0.7$ and $top\_p = 1$, with no penalties.

- We also employed models optimized for chat completion `GPT-3.5-turbo`, that we refer to as **GPT-3.5T**, and the most recent **GPT-4**. We used the chat completion API to generate synthetic instances, fitting the prompt to the chat structure but keeping the same layout and

strategies described previously for GPT-3.5.

The classifiers were implemented by fine-tuning `bert-base-portuguese-cased` (SOUZA; NOGUEIRA; LOTUFO, 2020). This model was trained with initial weights from BERTMultilingual and further pre-trained on brWaC (FILHO et al., 2018), a large Web corpus for Brazilian Portuguese with 145 million sentences and 2.7 billion tokens. The model is case-sensitive and has a vocabulary of 29,794 tokens. We used the same settings described in Section 4.1.3 and employed five times five-fold cross-validation, reporting the average results per class. Results were compared with Friedman statistical test (FRIEDMAN, 1937).

We compared the best results yielded by the classifiers trained with augmented data to two simple approaches of few-shot learning:

- We used **GPT-4 in zero-shot and few-shot** settings for classifying the test set of City-tweets. A simple prompt containing the instruction in Portuguese to "Classify a tweet into the following classes: Water, Transportation, Environment, *etc.* " was used. The list of the 16 classes was appended to the instruction. In the few-shot setting, we also appended one example of each class from the training set.
- We used the framework named **SetFit** (TUNSTALL et al., 2022) that facilitates the fine-tuning of a pre-trained ST model to small datasets in a contrastive Siamese manner: the batches are organized pairing the sentences: positive pairs (similar) contain sentences from the same class, and negative pairs have sentences from different classes. The resulting model is then used to generate embeddings, which are used to train a classification head. It is important to understand that the results of this process also depend on the quality of the pre-trained ST model used in the fine-tuning process. We utilized the multilingual ST model paraphrase-multilingual-mpnet-base-v2 (REIMERS; GUREVYCH, 2019), which Tunstall et al. (2022) reports good results in the English version. We used 20% of the training set as validation. We trained for one epoch with a learning rate of 2e-5, batch size of 32, AdamW as the optimizer, and cosine similarity as the distance metric used by the contrastive loss function.

4.2.3 Quantitative Analysis

In this section, we present the quantitative results considering three dimensions: the augmentation statistics (Section 4.2.3.1), the macro-F1 results of each classifier trained with original and augmented instances (Section 4.2.3.2), and an analysis of the processing times (Section 4.2.3.3).

Table 4.6 – Augmentation statistics for City-tweets.

	Instances	Sim.	Toks.	Len.
O	201		27	125
BT	565	0.90	27	139
EDA-BT	1481	0.82	26	138
EDA-pt	1290	0.87	26	137
CWE	1976	0.81	26	124
LMPG-BT	1590	0.81	23	115
GPT-3.5	1536	NA	27	154
GPT-3.5T	1536	NA	34	199
GPT-4	1536	NA	34	182

Note: The “O" row refers to the original instances. The remaining rows show the number of synthetic instances (S) produced by each DA method. “Sim” is the average cosine similarity between synthetic and original instances (GPT methods do not have the similarity score because they are based on all instances of the class and not only in an original base instance). “Toks” is the average number of tokens per instance, and “Len” is the average number of characters per instance.

4.2.3.1 Augmentation Statistics

The statistics for the DA methods on City-tweets are presented in Table 4.6. As a general tendency, the similarity in relation to the original instances was lower than what was found in most English low-data scenarios (Table 4.2). In LMPG-BT, the reduction in the number of tokens was less aggressive than observed in LMPG for the English low-data scenarios. We attribute this to the added variety introduced by running the BT step before and after paraphrase generation.

The GPT models have the same numbers in the “Instances” column because the prompting strategy requested all 128 choices, *i.e.,* we added 128 new synthetic instances for each class submitted to the augmentation procedures. The instances generated by GPT-3.5T and GPT-4 have, on average, 29% more tokens and are 23% longer than GPT-3.5. Since the synthetic instances in all GPT models were based on many original instances, calculating their similarity with the original base instance is not applicable.

4.2.3.2 Classification Quality

The classification results are presented in Table 4.7. GPT-3.5 yielded the highest score with 43% improvement over the baseline (without augmentation). However, this result is similar to the results obtained by BT, EDA-BT, and GPT-4. We compared the 25 models generated (five-fold cross-validation repeated five times) and found no statistical difference considering $\alpha = 0,05$. Among the winning methods, BT generated the fewest augmented instances. DA methods almost always improved the results in each class. The exceptions were BT, which did not improve the category Health, and CWE, which did not improve Economy and Health. In addition, we trained a classifier combining all augmented instances, removing eventual duplicates generated by the five

DA methods. The result was worse than when the methods were used in isolation (0.691), which shows that good results are not a direct consequence of quantity.

Table 4.7 – Classifier performance with DA methods in City-tweets. Each method has the total number of instances used in training (O+S) and the F1-score averaged for the five cross-validation runs. The top-scoring results are in bold.

		O		**BT**		**EDA-pt**		**EDA-BT**		**CWE**		**LMPG-BT**		**GPT-3.5**		**GPT-3.5T**		**GPT-4**	
Label	**Class name**	**#**	**F1**	**#**	**F1**	**#**	**F1**	**#**	**F1**	**#**	**F1**	**#**	**F1**	**#**	**F1**	**#**	**F1**	**#**	**F1**
0	Economy	19	0.629	63	0.777	202	0.581	153	0.780	132	0.759	163	0.735	147	**0.808**	147	0.728	147	0.784
1	Education	12	0.716	41	0.758	129	0.750	109	**0.845**	87	0.750	93	0.794	140	0.843	140	0.820	140	0.775
2	*Energy*	*67*	*0.909*	*67*	*0.929*	*67*	*0.935*	*67*	*0.935*	*67*	*0.934*	*67*	*0.926*	*67*	*0.920*	*67*	*0.913*	*67*	*0.927*
3	Enviroment	16	0.335	59	0.520	175	0.420	132	**0.546**	120	0.440	148	0.457	144	0.525	144	0.542	144	0.537
4	Finance	12	0.236	46	0.671	132	0.545	103	**0.784**	96	0.649	105	0.565	140	0.672	140	0.585	140	0.570
5	Fire/Emerg	14	0.323	56	0.696	149	0.675	116	0.612	105	0.744	122	0.492	142	**0.781**	142	0.616	142	0.715
6	Health	35	0.803	139	0.791	376	0.765	295	**0.877**	260	0.865	327	0.834	163	0.838	163	0.845	163	0.854
7	*None*	*670*	*0.862*	*670*	*0.863*	*670*	*0.851*	*670*	**0.867**	*670*	*0.864*	*670*	*0.855*	*670*	*0.865*	*670*	*0.862*	*670*	*0.866*
8	Recreation	13	0.023	52	0.550	143	0.486	111	0.581	104	0.549	119	**0.590**	141	0.484	141	0.385	141	0.415
9	Safety	32	0.355	116	**0.556**	346	0.423	257	0.380	217	0.375	276	0.535	160	0.495	160	0.490	160	0.533
10	Shelter	11	0.505	43	0.876	121	0.876	92	**0.971**	83	0.873	104	0.856	139	0.928	139	0.872	139	0.926
11	Solid Waste	19	0.555	81	0.631	207	0.668	165	0.669	150	0.696	162	**0.716**	147	0.709	147	0.692	147	0.660
12	Telecomm.	8	0.016	33	**0.626**	88	0.554	65	0.319	60	0.251	75	0.561	136	0.553	136	0.551	136	0.473
13	*Transportation*	*342*	*0.821*	*342*	*0.810*	*342*	*0.823*	*342*	*0.820*	*342*	*0.813*	*342*	*0.814*	*342*	*0.816*	*342*	*0.810*	*342*	*0.813*
14	Wastewater	10	0.357	37	0.743	109	0.746	84	0.757	77	0.768	97	0.759	138	0.830	138	**0.878**	138	0.837
15	*Water*	*111*	*0.920*	*111*	*0.976*	*111*	*0.963*	*111*	*0.975*	*111*	*0.971*	*111*	*0.973*	*111*	*0.967*	*111*	*0.956*	*111*	*0.970*
	Instances / Avg F1	1391	0.523	1956	**0.736**	3367	0.691	2872	**0.732**	2681	0.706	2981	0.716	2927	**0.752**	2927	0.721	2927	**0.728**

Table 4.8 – DA winners compared to Few-shot and Zero-shot approaches

Model Setting	F1-macro	Training cost	Inference cost
BT	0.736	NA	NA
EDA-BT	0.732	NA	NA
GPT-4 TG for DA	0.728	7.05	NA
GPT-3.5 TG for DA	**0.752**	2.11	NA
GPT-4 zero-shot	0.661	NA	3.8
GPT-4 few-shot	0.726	NA	17.9
SetFit MPNET MULTIL.	0.707	NA	NA

Note: costs are in USD. Costs at the time of writing this thesis for GPT-4 were 0.03 and 0.06 USD to input/output 1k tokens. GPT-3.5 (text-davincci-003) is no longer available, but the cost was 0.02 USD per 1k tokens when running the experiments.

The zero and few-shot learning results are presented in Table 4.8. The first four lines have the winning DA methods. For text generation (TG) with GPT-3.5 and GPT-4, the prompt per class strategy described in Section 4.2.2 with all examples was used. The training costs for those methods refer to the cost of generating the synthetic instances used to train the classifier, which is done only once. The inference costs for GPT-4 zero- and few-shot are related to the 600 instances in the test set. Thus, for the few-shot setting, we have around 0.03 USD per tweet to be classified. For the zero-shot setting, it is about 0.01 USD per tweet. Analyzing the costs and benefits of the few-shot and zero-shot approaches compared to the winning DA methods, we can say that, in the context of the City-tweets dataset, investing in augmenting the classes brings higher results. The costs involved in the GPT-4 few-shot approach, which yielded the closest results to the best DA methods, are per each tweet sent to the model for inference. The costs should be estimated when deciding to go live with this kind of solution. All those topics have to be considered when choosing the approach.

For the City-tweets dataset, established DA methods such as BT and EDA-BT are still valuable – they are among the best scoring in Tables 4.7 and 4.8, which are in the group of winners. For research purposes, we see that generating synthetic instances through a prompting strategy per class with a powerful language model, as was done here, yielded the highest result. Text generation with GPT-3.5+ models may also perform well to generate pretty new instances for a synthetic NLI, the primary goal of this thesis.

The few-shot approach fine-tuning a multilingual sentence-transformers (ST) model generated interesting results, although it was not one of the DA winning methods. The SetFit framework performs a type of augmentation since it combines the available training instances (single text passages with a label), assembling them as positive pairs from the same class and negative pairs when in different classes. At this point, we saw an opportunity to improve the ST model used by SetFit. Our goal is to create a synthetic NLI dataset, detailed in Chapter 5, and use it to

train an ST model that can produce good embeddings to represent text. We measured the results of this new ST model using the SetFit framework for standard classification (described in Chapter 6).

4.2.3.3 Processing times

Regarding the computational cost involved in the DA methods, GPT models depend on the OpenAI API infrastructure: there are rate limits[11] to the number of tokens and requests per minute. These restrictions vary according to the payment plan. However, they can also be affected by high demand during rush hours.

Considering the non-GPT methods, the fastest was EDA-pt, which ran in a matter of seconds. EDA-BT has a higher macro-F1 than EDA-pt, but it has an additional translation step, thus taking around 30 minutes for the full training set. BT is the most time-consuming due to the eight translations for each original instance (two translations per instance for the four languages, for example, from Portuguese to Arabic and from Arabic to Portuguese); therefore, it takes from three to four hours. The time spent on CWE is basically the inference time of the model that depends on the infrastructure available (CPU, GPU, memory, *etc.*). The same happens for LMPG-BT – the first cost to consider is the inference time of the model, but it has the overhead of requiring translation to use the Pegasus model in English.

4.2.4 Qualitative Analysis

To achieve the overall goal of this research and produce a synthetic NLI dataset, we need well-written instances. Hence, to assess the quality of the augmented instances generated by the different DA methods, including the generative methods, we took a random sample of 20% of the original instances and analyzed two augmented instances generated by each DA method. The GPT synthetic instances are generated based on all instances in one class. Therefore, no comparison with the original base instance was made since the calculation considers an average on the similarity of each augmented instance and its original base. We took the same number of GPT synthetic instances per class as we did for other methods. The analysis was based on the following four criteria.

- *Readability* (R). If the content is clear and understandable despite grammar errors.
- *Label Preservation* (L). If the augmented instance still belongs to the original label.
- *Semantic Preservation* (S). If the semantics is the same as in the original instance. There are cases in which the label is preserved, but the semantics of the sentence is different. For example, two instances may belong to the same label, *e.g.* “Recreation”, but they can be

[11] https://platform.openai.com/docs/guides/rate-limits/overview

semantically different since they refer to different places: *"we love spending time on the Guaíba lake"*, and *"we love spending time in the Redemption Park"*. For the GPT models, this analysis is not applicable.

- *Correctness* (C). If there are **no** grammar errors or typos introduced in the augmented instances.

The qualitative evaluation results are presented in Table 4.9. Synthetic instances are, in general, readable. All methods achieve a good label preservation rate (above 70%). BT and EDA-pt are the non-GPT methods that preserved most labels, while CWE had the lowest label preservation score. We did not find a correlation between label preservation and classification quality. EDA-pt, for example, has the second-best label preservation rate and the worst F1 score. It is worth noticing that CWE sometimes replaced words that bring different meanings (for example, "se joga tudo que é *lixo* na rua" and "se joga tudo que é *roupa* na rua", which can be translated as "everything that is *garbage* is thrown on the street" and "everything that is *clothes* is thrown on the street") which causes the degradation of label preservation since for the second example is not clear that is talking about solid waste.

BT and LMPG-BT are the methods that introduce the fewest errors with a high (C) score, while EDA-pt is the method that introduces the most errors. We could not identify a correlation between the presence of errors and a decrease in performance; for example, EDA-BT had the same (C) score as CWE, but the former is one of the winning methods, while the latter had the lowest results among all methods. The synthetic instances generated by the GPT methods are highly readable and grammatically correct, indicating a good potential to be used to create the NLI synthetic dataset, the main goal of this thesis.

An example of an original instance and its augmented versions generated by the different DA methods is in Table 4.10. In EDA-pt, there is an error duplicating the expression "microwave oven", but this did not interfere with sentence comprehension, label, or semantics. We found two changes in the label and three changes in semantics in the following ways:

- changes of intent: the original instance wishes to find out where to discard a microwave oven, but some augmented instances change the intent by asking the question "can I discard?" or affirm "I do not know how to discard".
- changes in the object: a charcoal oven is not the same as a microwave oven, and their discarding processes are different.
- changes in the label: in EDA-BT example, the word "expulsar" *kick out* in Portuguese is not a synonym for discarding trash. The label here is "Solid Waste", which is clearly associated with discarding but not with kicking out. In the CWE example, the word "utilizar" *using* in Portuguese removes the clarity that the sentence is about "Solid Waste".

Table 4.9 – Qualitative analysis of DA methods applied to City-tweets.

	R	L	S	C
BT	99%	95%	65%	90%
EDA-BT	95%	86%	26%	65%
EDA-pt	95%	94%	53%	50%
CWE	95%	74%	13%	65%
LMPG-BT	99%	81%	33%	90%
GPT-3.5	100%	100%	NA	100%
GPT-3.5T	100%	100%	NA	100%
GPT-4	100%	100%	NA	100%

Note: It was considered Readability (R), Label preservation (L), Semantic preservation (S), and Correctness (C) of the synthetic instances

Table 4.10 – Example of an original instance and its augmented versions generated by the different DA methods.

		R	L	S	C
Original	**Onde posso descartar um forno de microondas?** ***Where can I discard a microwave oven?***				
BT	Onde posso descartar um forno de micro-ondas? *Where can I discard a micro-wave oven?*	✓	✓	✓	✓
CWE	Onde posso utilizar um forno de carvão? *Where can I use a charcoal oven?*	✓	✗	✗	✓
EDA-BT	onde posso expulsar de um forno de microondas. *where can i kick out a microwave oven.*	✓	✗	✗	✓
EDA-pt	onde posso descartar forno de micro ondas um forno de microondas. *where can i discard microwave oven a microwave oven.*	✓	✓	✓	✗
LMPG-BT	Eu não sei onde descartar um forno de microondas. *I don't know where to discard a microwave oven.*	✓	✓	✗	✓
GPT-3.5	Uma cidade limpa é uma cidade que se cuida. Não joguem lixo na rua e contribuam para o meio ambiente. #ResíduosSólidos #CidadeLimpa. *A clean city is a city that takes care of itself. Do not throw garbage in the street and contribute to the environment. #Solid-Waste CleanCity*	✓	✓	na	✓
GPT-3.5T	Não é opção jogar lixo na rua, é obrigação de cada um de nós cuidar do nosso planeta. #resíduossólidos #meioambiente #cidadania #compromisso *It is not an option to throw garbage in the street, it is the obligation of each one of us to take care of our planet. #solid waste #environment #citizenship #commitment*	✓	✓	na	✓
GPT-4	Odeio ver pessoas jogando lixo na rua como se fossem animais sem consciência! Acordem, seu lixo é sua responsabilidade! Estamos todos neste planeta juntos. Consciência, por favor! *I hate to see people throwing garbage in the street like animals without conscience! Wake up, your trash is your responsibility! We are all on this planet together. Conscience, please!*	✓	✓	na	✓

Note: Our data is in Portuguese, and we provide English translations to allow comprehension by a wider audience.

4.3 Summary and Discussion

In this chapter, we investigated the application of various DA techniques to aid in supervised text classification. The purpose of those experiments was to find the means to handle very small datasets and classes and observe as text generation behaves along with the other DA methods, looking to pave our knowledge to be able to produce the overall goal of this thesis: the synthetic NLI in Portuguese, which is detailed in next Chapter. Following are the answers to the research questions this chapter strives to respond to:

RQ1. Which augmentation strategies should be selected for text classification to pave the understanding of generative approaches to produce synthetic instances?
We looked for methods still present in recent papers (or because researchers have adapted or used them as a baseline). Text generation methods received special attention. We also prioritized techniques that are effortless to implement and appear straightforward to adapt to Portuguese.

RQ2. Besides improving the final classification results, what else is important in DA to ensure high-quality synthetic instances?
We evaluated the classification results using macro-F1 and compared the methods. In Portuguese, we analyzed the performance within the 16 classes. Besides this, we ran zero-shot and few-shot experiments to understand whether the augmentation approaches, including the generative ones, were competitive with the few-shot results, and they are, in the context of this use case.
Finally, we run a qualitative analysis over 20% of the augmented instances in Portuguese, using the dimensions of Readability, Label Preservation, Semantic Preservation, and Correctness. We could observe that text generation with GPT-3.5+ models delivered highly readable instances without errors.
We could discuss if errors would not be desirable to the specific use case (tweets), but we see in the macro-F1 results that among the winners, we have GPTs methods with perfect writing, but also EDA-BT, which introduces errors (65% of correctness), reminding that with EDA words are removed/inserted/swapped several times corrupting the quality of the sentence. This thesis does not have the purpose of evaluating if errors or noise as DA would improve or decrease the classification results. We could have applied and analyzed several methods in this context (add a large set of emojis, hashtags, or any kind of noise, such as corrupting words, for example, that would modify the original sentence). Still, we prefer to use methods not only applied to tweets but also those found in the papers of our exploratory search (see response of RQ1). Additionally, we do not want errors when generating pairs of premise and hypothesis in the synthetic NLI dataset, discussed on Chapter 5.

We have not evaluated the creativity of the methods. However, when we see high label preservation combined with low semantic preservation, we can interpret it as a good indicator of variability. Since we are interested in the quality of GPT instances, we used class descriptions and all examples to generate new sentences. We manually inspected instances, and they were very creative.

RQ3. Is text generation competitive for DA? What are the pros/cons compared to other methods? What insights acquired exploring DA can be investigated in the next step to produce synthetic instances for NLI in Portuguese?

Among the four winning methods, two were based on text generation. As mentioned in the RQ2 answer, the quality of the writing is superior, and they could produce creative answers.

The high-quality text generated by the GPT models for Portuguese exceeded our expectations since the original GPT3 training data contains 93% of English words and only 7% of words in other languages. Portuguese represents only 0.52% of the entire dataset[12].

In *InstructGPT*, more than 96% of the training data are in English. Still, authors report promising qualitative findings that the model can sometimes follow instructions in other languages, but they have not tracked this behavior quantitatively. The suggestion to explain this behavior stands that alignment methods could generalize to produce the desired behavior on inputs that humans did not directly supervise (OUYANG et al., 2022). Large language models specialized in following instructions, such as GPT models, are very promising options to produce synthetic instances and be applied for DA. Thus, instead of using these models to perform the downstream task, which would incur costs at inference time for every test instance (or instances used on a service that goes live), one could use them to generate good and diverse training data.

On the cons side, we have the cost. It requires careful calculation, forecasting costs, and designing the right strategy before running a long batch. Counting the tokens and estimating output lengths are the ways to do that. However, the API costs may be lower compared to the costs of human writing examples and labeling. Another point to pay attention to is the rate limits[13] of the API. The use of the services is straightforward. However, depending on the desired volume of tokens in the input and output, one can reach the limit, but we could manage it without problems.

Analyzing how to use the knowledge acquired in those DA experiments to generate the NLI dataset, we noticed that each part of the context used (the prompt is a combination of the class description, the instruction, and the examples added in a few-shot settings) influences the result. Crafting different prompts and varying those components appears as a strategy for developing the synthetic instances for NLI. We figured out that we needed to attempt some prompts to understand if we should have gone for a single generation process, similar to what we did when generating

[12]GPT3_statistics:<https://github.com/openai/gpt-3/blob/master/dataset_statistics/languages_by_word_count.csv>

[13]https://platform.openai.com/docs/guides/rate-limits/rate-limits

several instances based on one prompt per class in the Portuguese use case. Or, if we split the process generating the premises first and, based on that output, a second process would generate the hypothesis. As found in Chapter 5, we used a two-stage generation to produce the NLI instances.

Based on the answers above, we conclude that the DA experiments allowed a comparison between conventional and generative techniques. The main goal of this comparison was to leverage the knowledge for the next step: adapt the TG scripts to the NLI synthetic instances. The contributions delivered in this chapter are as follows:

C1. An analysis of the impact of DA methods over simulated low-data scenarios using well-known public text classification datasets in English.

C2. A dataset named City-tweets with 1993 tweets in Portuguese labeled with the Smart City dimension they refer to.

C3. The adaptation of DA techniques that were originally designed for English classification tasks to work with Portuguese.

C4. A comparison of the DA techniques in a multiclass single-label classification problem with a Portuguese dataset both in quantitative and qualitative terms.

5 SYNTHETIC INSTANCES FOR NLI

In this chapter, we address the following research question:

RQ4. How to generate a synthetic dataset for NLI in Portuguese? How to evaluate the results?

We took advantage of the learning acquired on text generation with GPT in the context of DA for textual classification described in Chapter 4. Aiming to contribute to language resources for Portuguese, this chapter introduces InferBR, a Portuguese NLI dataset semi-automatically generated using the GPT-4 but with humans revising the synthetic instances. The generation process was conducted separately, first to generate the premises and secondly the hypotheses.

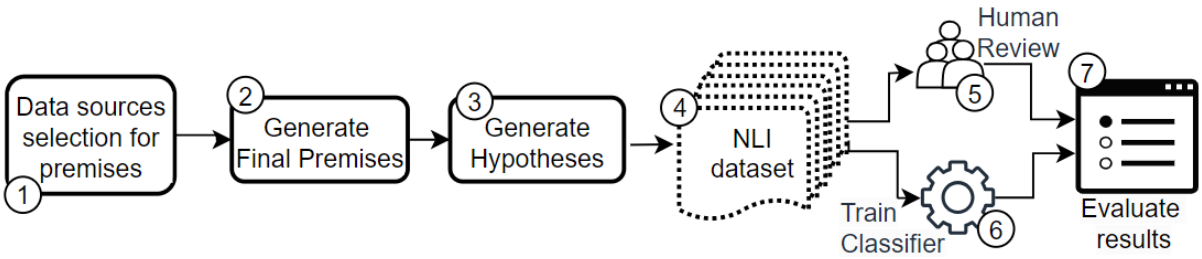
The high-level generation process is presented in Figure 5.1 with the following steps: selection of the data sources that provide examples to be used to generate the premises (1), generation of the final version of the premises (2), generation of the hypotheses derived from each premise (3), resulting in the dataset (4) which is submitted to human review (5). A classifier is trained (6), and, in the end, the evaluation is conducted by summarizing human validation findings and evaluating inter-dataset experiment results from crossing models and test sets with the other two existing NLI datasets in Portuguese (7).

We used two datasets as sources to produce the premises and to design prompts submitted to GPT-4 to generate the three types of NLI hypotheses. The generated dataset was manually revised. In all generative tasks, we set the temperature to 0.8 for decoding. The cost of using GPT-4 via the Open API was USD 220. This amount also includes trials that were not used in the final dataset. The next sections describe in detail the processes to generate premises and hypotheses.

5.1 Generating Premises

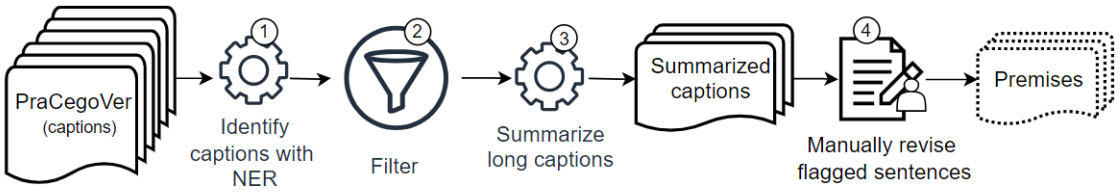
We want to generate premises that are comprehensible, unambiguous (any person would understand the same from it), informative (do not need additional data to understand it), coher-

Figure 5.1 – Macro-processes to generate the NLI dataset.

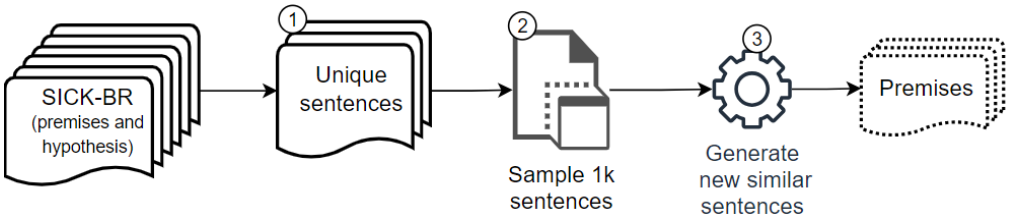


Source: the author

Figure 5.2 – Premise generation process.



(a) Process A



(b) Process B

Source: the author

ent (logically structured, easy to read), and that respect common sense. We used two datasets as sources for the premises: PraCegoVer (SANTOS; COLOMBINI; AVILA, 2022) and SICK-BR (REAL et al., 2018). The process used to get the final premises from each dataset was different, as shown in Figure 5.2. PraCegoVer is an image-caption dataset where captions are used to extract raw premises that are processed as described in Section 5.1.1 SICK-BR is an NLI dataset. We use some of its sentences to serve as the context to generate new premises, as detailed in Section 5.1.2.

5.1.1 PraCegoVer

PraCegoVer (SANTOS; COLOMBINI; AVILA, 2022) is a multi-modal dataset with images and their descriptions in Portuguese (captions). It came from a social movement that started in 2012, aiming to increase the inclusion of people with visual impairments, and its goal was to contribute to developing models to automate image captioning. The initiative stimulated social Instagram users to post images tagged with #PraCegoVer and add a short description of their content. The image captions have 39.3 words on average, and the standard deviation is 29.7, this is considered very challenging for image captioning tasks compared to other datasets such as Flickr30k (YOUNG et al., 2014b) and MS-COCO (LIN et al., 2014), which have, on average, ten words per caption and low variance.

This dataset contains many named entities, including several brand names. This is because companies used the #PraCegoVer to post messages describing images of their products. Even people's names are in the dataset because users would post photos with friends and include their

Table 5.1 – Examples of the summarization step.

01) **O**: As Fotos mostram o espaço do bicicletário por diversos ângulos. É possível os espaços onde as bikes devem ser alocadas. *The photos show the bike rack from different angles. Spaces where bikes should be allocated are possible.* **G**: O bicicletário é exibido de vários ângulos, destacando os locais designados para a alocação das bicicletas *The bike rack is displayed from multiple angles, highlighting designated bike storage locations*
02) **O**: Foto em preto e branco com imagem do apresentador no palco, com microfone na mão. *Black and white photo with image of the presenter on stage, with microphone in hand.* **G**: O apresentador está no palco segurando um microfone *The presenter is on stage holding a microphone*
03) **O**: foto de porquinhos recém nascidos, de olhos fechados e a frase "Somos só bebês" *photo of newborn piglets, with their eyes closed and the phrase "We are just babies"* **G**: Recém-nascidos porquinhos de olhos fechados são identificados como apenas bebês *Newborn piglets with their eyes closed are identified as just babies*

Note: (O) corresponds to the original image caption; (G) is the generated summary. We also show the translation into English in italics.

names in the descriptions.

The process for generating the premises using PraCegoVer is depicted in Figure 5.2a. To filter out descriptions with proper nouns, in Step (1), we run a model trained for Named Entity Recognition (NER)[1]. Instances containing entity names are flagged and then removed in Step (2), where we also remove instances with special characters. In Step (3), we used GPT-4 to summarize the filtered image captions since they contain long descriptions with several details, and we wanted to reduce them to obtain a concise premise. The instruction sent to the model requested a summary of the scene, avoiding mentioning it is about a picture (since several descriptions have those mentions when describing the image). Some examples of the original caption (O) and the generated summary (G) can be seen in Table 5.1. It is worth pointing out that some captions in PraCegoVer have grammar and orthography errors – that is the case of example 01 in which there is a missing verb between "*possível*" and "*os*". These issues end up being corrected by the summarization step. Finally, in Step (4), we revise premises with the following issues: errors pointed out in a spell-checker or containing some expressions that could make the premise confusing and that are repeated in several descriptions like *"retratado"* ("*portrayed*", in English), *etc.* During this phase, around 25% of the generated premises were flagged to be revised, and almost 60% of this group was modified to yield coherent text.

[1]<https://huggingface.co/monilouise/ner_news_portuguese>

Table 5.2 – Premises generated from sample sentences in SICK-BR.

O: Muitas crianças estão de pé *Many children are standing* **G**: Vários jovens estão sentados *Several young people are sitting*
O: Não tem nenhum cachorro perseguindo uma bola *There is no dog chasing a ball* **G**: Não há nenhum gato brincando com o novelo de lã *There is no cat playing with the ball of yarn*
O: Uma pessoa está imprudentemente montando um cavalo *A person is recklessly riding a horse* **G**: Um indivíduo está temerariamente conduzindo um automóvel *An individual is recklessly driving a car*
O: Quatro pessoas estão paradas silenciosamente ao ar livre *Four people are standing silently outdoors* **G**: Três indivíduos permanecem quietos sob a luz da lua *Three individuals remain quiet under the moonlight*

Note: (O) corresponds to the original sentence, and (G) is the sentence generated by GPT-4. In italics, the translation to English.

5.1.2 SICK-BR

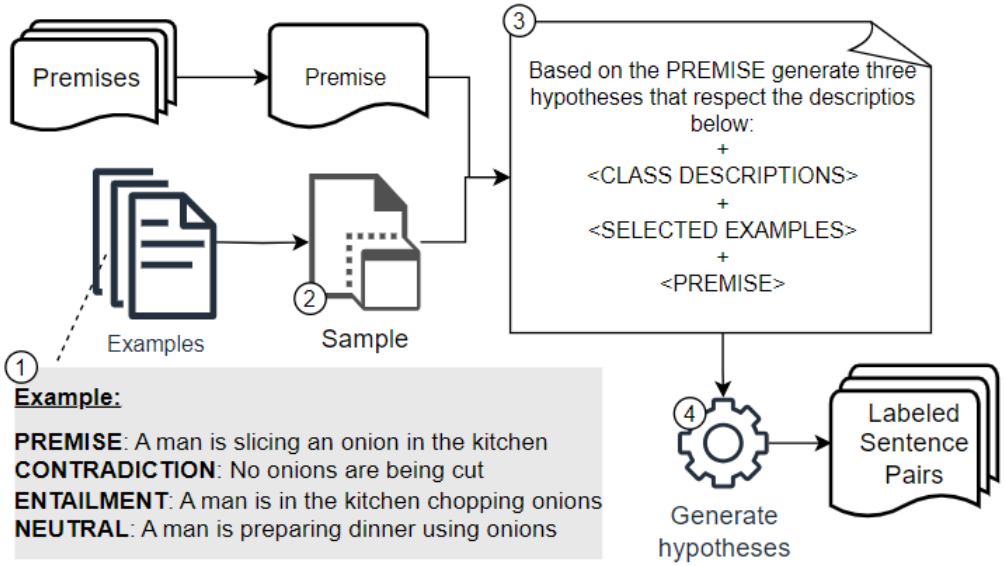
SICK-BR (REAL et al., 2018) originated from the translation of the English dataset SICK (MARELLI et al., 2014). Figure 5.2b presents the process of generating premises from SICK-BR. In Step (1), we sample unique sentences from the union of premises and hypotheses from the training and validation splits. There are 4.5k unique premises and 4.6k unique hypotheses. A total of 3.2k sentences appear both as a premise and hypothesis, but they are paired with different combinations to make up unique pairs. Considering premises and hypotheses jointly, SICK-BR has around 6k unique sentences in training and validation sets. From this final set of 6k, in Step (2), we sample 1k sentences (16% of the unique sentences) to be used as sources to generate our premises. In Step (3), we prompt GPT-4 with the 1k sentences and request the model to write new instances with similar contexts. This instruction ensures semantic variations that allow more diversity than just paraphrasing. Table 5.2 presents some examples of the source and the generated instances.

5.2 Generating Hypotheses

We used a few-shot inference strategy, depicted in Figure 5.3. We first write a list of 50 examples considered representative and challenging. Those 50 examples are randomly used in the prompt. Each example corresponds to the layout indicated in number (1) of the flow. In step (2),

for each premise, we sample three examples from the 50 available and add them to the prompt. The examples used for each premise may vary, which is desirable to add diversity to the dataset. Step (3) presents the structure of the prompt, which has four components: (i) an initial request – the same for all premises, (ii) short descriptions for the three classes, (iii) the three selected examples, and (iv) the premise for which the hypotheses should be generated. In Step (4), we used GPT-4 to generate the hypotheses. The output is three labeled hypotheses that, along with the premise, compose the three labeled pairs for the dataset.

Figure 5.3 – Hipotheses generation process



Source: the author

5.3 Results

This section first presents the synthetically generated dataset named InferBR, which is detailed in Section 5.3.1. Secondly, in Section 5.3.2, we present the human validation process and results of revising all pairs. In closing this Section, in Section 5.3.3, we compare the performance of models trained with the generated data of InferBR in recognizing NLI classes in the other two datasets in Portuguese (ASSIN2 and SICK-BR) and vice versa.

Table 5.3 – Number of instances per split and class for the NLI datasets in Portuguese

	train	val	test
InferBR			
Contradiction	2,800	215	586
Entailment	2,799	216	586
Neutral	2,800	215	586
Total: 10,803			
SICK-BR			
Contradiction	998	224	202
Entailment	1,948	437	436
Neutral	3,941	815	839
Total: 9,840			
ASSIN2			
Non-entailment	3,250	250	1,224
Entailment	3,250	250	1,224
Total: 9,448			

5.3.1 The InferBR dataset

InferBR was created using two strategies to generate the premises: 41% of the data was generated using process B with SICK-BR premises as the source to create new instances with a similar context but more aggressive than paraphrasing to yield more diversity. Most of the data, 59% , came from PraCegoVer using Process A. The rationale was to have more instances that were not present in the other two NLI datasets. We generated the sets for training and testing, and Table 5.3 presents the numbers per split and class for InferBR and the other two existing NLI datasets in Portuguese. In our dataset, the premises in training and validation sets do not appear as premises in the test set.

The premises in InferBR were longer compared to SICK-BR, as can be seen in Figure 5.4, and the generated hypotheses were shorter than the premises. This reduction of the hypotheses also occurs in the English benchmark SNLI (BOWMAN et al., 2015a).

To evaluate possible overlaps among the three datasets ASSIN2, SICK-BR, and InferBR, we checked for duplication of premises and hypotheses both within the individual datasets and across datasets. The results are in Table 5.4. InferBR has fewer unique premises (P) but many more than twice the number of unique hypotheses (H). In addition, there is almost no duplication when analyzing unique sentences in the dataset ($P \cup H$). We see this as an advantage since we ensure more diversity to the hypotheses and also may allow us to apply future augmentation techniques to combine similar H from this larger set in different ways, improving the learning process. It is worth pointing out that although InferBR used SICK-BR as a source for premises and hypotheses, the resulting dataset has a very small overlap with SICK-BR (8 and 4, respectively). This happens because our process can change the original context.

Table 5.4 shows that, as described in the original paper, most of the sentences in ASSIN2

Figure 5.4 – Distribution of tokens

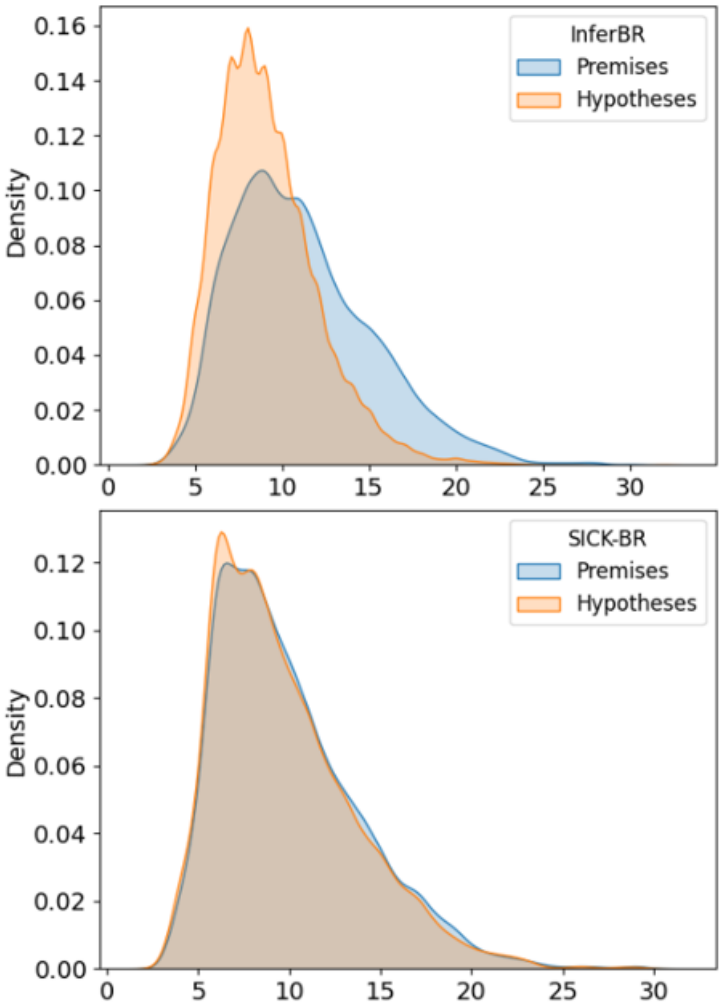


Table 5.4 – Unique sentences.

Set	P	H	$P \cap H$	$P \cup H$
ASSIN2	5,150	5,172	3,814	6,508
SICK-BR	5,001	4,929	3,846	6,084
InferBR	3,600	10,669	450	13,819
ASSIN2 ∩ SICK-BR	3,901	3,794	2,642	5,053
InferBR ∩ SICK-BR	8	4	1	11
InferBR ∩ ASSIN2	10	4	2	12

Note: Unique premises (P), hypotheses (H), sentences ($P \cup H$), and their intersection ($P \cap H$) calculated intra- and inter datasets.

Table 5.5 – Average occurrence of each POS class in premises and hypotheses.

POS	Premise		Hypothesis	
	InferBR	**SICK-BR**	**InferBR**	**SICK-BR**
adjective	0.79	0.60	0.58	0.60
adposition	1.80	1.46	1.28	1.45
adverb	0.23	0.32	0.24	0.30
auxiliary	0.85	1.08	0.78	1.08
coord.conj.	0.24	0.23	0.11	0.24
determiner	1.84	1.84	1.54	1.84
noun	3.43	2.97	2.73	2.96
numeral	0.13	0.12	0.05	0.12
pronoun	0.14	0.11	0.12	0.11
proper noun	0.09	0.06	0.06	0.06
punctuation	0.23	0.05	0.52	0.05
subord.conj.	0.08	0.01	0.06	0.01
verb	1.31	1.24	1.09	1.23

come from SICK-BR. The pairs in ASSIN2 were reorganized to ensure the balance between the two classes (entailment and non-entailment), different from SICK-BR, which has three classes but is imbalanced.

To analyze the Part-Of-Speech (POS) categories in the datasets, we ran a POS tagger from *spaCy*[2] using the largest model available `pt_core_news_lg`. For each category, we calculated the average occurrence in the premises and hypotheses in InferBR and SICK-BR.

The results are presented in Table 5.5, where we can see that InferBR has more adjectives in the premises. This is expected because PraCegoVer instances refer to many brands' advertisements and people expressing opinions on the social network when describing the image ("*beautiful t-shirt*", "*gentle body lotion*"). We also see more nouns in InferBR, which is naturally related to the longer sentences, but also because it presents more detailed descriptions, for example, dishes ("*shrimp with tomatoes and grilled okra*") or what people are wearing ("*The person is getting ready to go out in her red bodysuit, striped pants, and red sandals*").

Both premises and hypotheses have a lower incidence of auxiliary verbs in InferBR. The occurrence of punctuation is higher in InferBR because it maintains the period in many instances, and more cases have a comma: "In the drinks cabinet, there are three bottles with labels.". InferBR's hypotheses have fewer determiners. This is associated with more cases where the article was replaced by the number ("*two girls are smiling*"), to a higher number of existential constructions (MCNALLY, 2021) ("*There are people at the geological site*."), and also to more presence of mass nouns ("*Water is coming out of the beach shower*").

Table 5.6 presents some examples of InferBR. The premises (P) are in a gray background, and below each premise line are the three hypotheses with respect to it.

[2]https://spacy.io

Table 5.6 – Examples of InferBR

A área rural tem plantações e agricultores trabalhando	P
A área rural é completamente desabitada.	C
Existem agricultores na área rural.	E
A área rural tem muitos pássaros.	N
A empresa anunciou que haverá um período de inatividade nas próximas duas semanas	P
A empresa continuará funcionando normalmente nas próximas duas semanas.	C
A empresa ficará parada nas próximas duas semanas.	E
A empresa terá uma reunião de planejamento na próxima semana.	N
Bisnagas de cor creme estão dispostas na pia	P
Não há bisnagas na pia	C
Há bisnagas, possivelmente de um produto específico, dispostas na pia	E
As bisnagas de cor creme contêm sabonete líquido	N
Foliões estão celebrando em uma avenida da capital	P
Foliões estão celebrando em uma cidade do interior	C
Está acontecendo uma celebração em uma avenida da capital	E
Foliões estão dançando samba	N
Não existe nenhum garoto com pintas azuis e verdes no rosto	P
Há um garoto com pintas azuis e verdes no rosto.	C
Não há garoto com marcas azuis e verdes na face.	E
Há um garoto com pintas vermelhas e pretas no rosto.	N
Um indivíduo está afagando as penas de um pássaro	P
Ninguém está tocando as penas de um pássaro	C
Alguém está tocando as penas de um pássaro	E
Uma pessoa está segurando um pássaro	N
Uma criança com um lenço bege na cabeça tem uma expressão séria	P
A criança está sorrindo.	C
A criança com um lenço bege está com uma expressão séria	E
A criança está segurando um bichinho de pelúcia.	N
A temperatura está caindo abaixo de zero graus	P
A temperatura está subindo acima de zero graus	C
A temperatura está se tornando negativa	E
A temperatura está influenciando o uso de aquecedores no ambiente	N
Um garoto de calça jeans está pulando	P
O garoto de calça jeans está sentado.	C
Um indivíduo vestindo calça jeans está em movimento.	E
O garoto de calça jeans está pulando em um trampolim.	N
Três indivíduos permanecem quietos sob a luz da lua	P
Todos os três indivíduos estão fazendo barulho	C
Sob a luz da lua, três pessoas estão em silêncio	E
Três indivíduos estão jogando futebol ao luar	N
Sopa está sendo consumida pela mulher	P
A mulher está jejuando	C
A mulher está se alimentando	E
A mulher está comendo sopa de legumes	N

Note: Each premise (P) is followed by three hypotheses: Contradiction (C), Entailment (E) and Neutral (N).

Table 5.7 – Statistics on the manual validation

Cohen's Kappa
Between R1 and R2: 0.9693
Between R1 and G: 0.9820
Between R2 and G: 0.9842
Validation Statistics
Number of pairs validated: 10,803
Agreements: 10,538
Dubious cases checked by R3: 93
Flagged as low quality: 275
Agreements (errors in the generated label): 10
Disagreements: 184
Confusing text: 81
Premises from Process A: 51
Premises from Process B: 15
Hypotheses: 15

Note: reviewers (R), the generated text and label (G).

5.3.2 Human Validation

Two MSc students (R1 and R2) with ongoing research in the NLP field evaluated all the pairs and labels. They were unaware of the processes that generated the premises and hypotheses, and they also did not access each other's annotations while the revision was ongoing. They annotated the label they understood as correct for each pair and judged if the text of premises and hypotheses was comprehensible, clear, unambiguous, coherent, and according to common sense. If the text did not have all these characteristics, they flagged it as "confusing". When reviewers were unsure about the label, they were oriented to flag the instance. Whenever an instance was flagged by at least one reviewer, it was checked by a third reviewer, a doctoral student (R3), who also assigned labels to these instances.

Table 5.7 summarizes the manual validation results. There is a very high agreement between reviewers and an even higher agreement between each reviewer's annotated label and the automatically generated one. Among the instances in which reviewers agreed and did not find any issue with the text, most of the labels assigned during the hypothesis generation process are correct (99.9%). The ten errors found are related to neutral boundaries with entailment and contradictions.

Only 2.6% of the instances were flagged as low-quality: 70% of those were hypotheses where the reviewers disagreed on the label, and no instances were flagged for R3 to check. Most of these instances are related to unclear boundaries between *Neutral* and the other two classes. 30% of the low-quality pairs are premises or hypotheses that were confusing for at least one reviewer and R3.

In total, only 0.75% of the pairs contain confusing text. It happens more in the premises

Table 5.8 – Examples of instances with confusing text and the investigated reason.

01)
P: O gato está miando no megafone
The cat is meowing into the megaphone
Source of error: premise generation process B using sentence (O) from SICK-BR:
O: O papagaio está falando no microfone
The parrot is speaking into the microphone
02)
P: Um homem e uma mulher estão um de frente
para o outro, com pássaros circulando as ramificações que emergem de suas cabeças.
A man and a woman face each other, with birds circling the branches that emerge from their heads.
Source of error: Not a real image
O: Casal, homem e mulher, um de frente ao outro. De suas cabeças saem ramificações com pássaros em volta.
Couple, man and woman, facing each other. Branches come out from their heads with birds around them.
03)
H: O campo florido está vazio de pessoas correndo.
The flower field is empty of people running.
Source of error: The entailment hypothesis is not well-written based on a well-written premise.
O: Não há ninguém correndo livremente em um campo florido.
There's no one running free in a field of flowers

Note: Premise (P), Original sentence (O), Hypothesis (H). In italics, the sentences are translated into English.

because a problem in the premise affects three instances with hypotheses derived from it. The root cause of the problem can be in processes A or B. Confusing text generated by process A is mainly related to describing unreal images or specific parts of a bigger scene (Example 02 in Table 5.8). On the other hand, the confusing premises generated by Process B are mostly cases where GPT-4 faced issues with common sense or world knowledge (Example 01: parrots usually imitate the human voice, and one could picture it in a microphone, but the cat meowing on the megaphone is not a good similar context choice). The confusing hypotheses are also related to GPT-4. Besides common sense issues, there are some problems with the fluidity of the text (in Example 03, the passage "*empty of people running*" is awkward, it would be better to write "*no one is running in the flower field*").

5.3.3 Comparing Classification Models

We trained NLI classifiers using the train and validation sets of InferBR, SickBR, and ASSIN2. The resulting classification models were used to predict instances in three settings: (i)

Table 5.9 – NLI Classification results with three classes

	Train	Test	Acc	F1ma	C-F1	E-F1	N-F1
1	SICK-BR	SICK-BR	.85	.85	.86	.81	.87
2	InferBR	InferBR	.90	.90	.89	.91	.90
3	SICK-BR	InferBR	.60	.58	.46	.64	.64
4	InferBR	SICK-BR	.64	.63	.54	.71	.65
5	SICK-BR	InferBR*	.59	.57	.44	.64	.64
6	InferBR	SICK-BR*	.65	.64	.56	.70	.65

Note: (C)ontradiction, (E)ntailment, and (N)eutral.
The symbol * means prediction over the entire dataset.

intra-dataset – the predictions are made on the test set of the original dataset; (ii) *inter-dataset* – the predictions are made on the test set of a different dataset; and (iii) *inter*-dataset* – the predictions are made on a different dataset, considering the full set of instances. The goal of the *inter* scenarios is to test the generalization power of the models trained on each dataset. The idea is to check if InferBR has an acceptable performance over the other datasets and vice-versa. A good performance in the inter-dataset scenario may indicate the model can recognize the classes in the other dataset. To do that, the models are presented with sentences that were not previously seen and that may come from a different domain. With that in mind, we did not run the model trained on SICK-BR to predict ASSIN2 labels, and vice-versa, because the overlap of the two datasets is very high, as can be seen in Table 5.4.

For each dataset, we fine-tuned the Portuguese model BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020) for eight epochs using early stopping criteria (if the validation loss stops decreasing after three steps – each step was configured to be half an epoch). We kept the same hyperparameters in all models, namely a learning rate of 3e-5, dropout of 0.1, and used the AdamW optimizer (LOSHCHILOV; HUTTER, 2019).

Table 5.9 presents the average classification results in terms of Accuracy and macro-F1 averaged across ten runs with different seeds. Each trained model was tested on the three scenarios. The intra-dataset performance (lines 1 and 2), as expected, always outperforms the inter scenarios. Comparing lines 3 and 4, we notice that the model trained on InferBR generalized better to SICK-BR than the reverse (line 4 has higher accuracy and macro-F1 than line 3). The same pattern repeats in lines 5 and 6. Analyzing the averaged F1-score for each class, we see that models have difficulties recognizing contradictions in the inter-dataset settings (*i.e.,* the scores in column C-F1 are always lower in lines 3 to 6 compared to lines 1 and 2).

We also ran experiments with two classes (entailment and non-entailment). We transformed the contradiction and neutral instances from SICK-BR and InferBR to non-entailment. The results are in Table 5.10. Again, InferBR achieved superior accuracy and macro-F1 in all inter-dataset scenarios (lines 4 to 11).

Table 5.10 – NLI classification results with two classes

	Train	Test	Acc	F1ma	NE-F1	E-F1
1	ASSIN	ASSIN	.87	.87	.86	.88
2	SICK-BR	SICK-BR	.88	.86	.91	.80
3	InferBR	InferBR	.93	.92	.95	.90
4	ASSIN	InferBR	.76	.72	.83	.62
5	InferBR	ASSIN	.82	.82	.82	.82
6	ASSIN	InferBR*	.77	.73	.83	.62
7	InferBR	ASSIN*	.80	.80	.79	.80
8	SICK-BR	InferBR	.78	.72	.85	.60
9	InferBR	SICK-BR	.79	.76	.84	.68
10	SICK-BR	InferBR*	.78	.72	.85	.58
11	InferBR	SICK-BR*	.79	.76	.84	.67

Note: Average classification results from ten models for each dataset with two classes: entailment (E) and Not-Entailment (NE). The symbol * means prediction over the entire dataset.

5.4 Summary and Discussion

In this chapter, we detailed the production of InferBR, an NLI dataset for Portuguese, proposing a semiautomatic process to generate premises and an automatic process to generate hypotheses exploring the text generation capabilities of GPT-4. The proposed processes can be easily adapted to other languages and tasks.

Following are the answers to the research questions this chapter addressed:

RQ4. How to generate a synthetic dataset for NLI in Portuguese? How to evaluate the results?

We developed two groups of processes, one to generate the premise and the second to generate the hypotheses.

We used a challenging dataset in Portuguese with image captions, PraCegoVer, with single detailed descriptions per image. We expected to bring common life descriptions when selecting this dataset. However, it has some advertising images, sometimes unreal, that can produce confusing premises. It was also necessary to minimize the occurrence of named entities and run an automatic process to flag the premises with specific characteristics (spelling errors, specific words) that were manually revised, which was around 25% of all premises originated from this dataset. This was the only manual activity to generate the dataset. Considering the total 13,819 of unique sentences (see $P \cup H$, in Table 5.4), 15% were manually revised. We also innovated on generating the premises using only 16% of an existing NLI dataset, SICK-BR, as initial context and producing very different sentences within a similar context.

We used language models for all the automatic transformations: summarization, named entity

removal, and text generation. For the generative tasks, we used the paid API of OpenAI using the GPT-4 chat endpoint. From the total amount spent, 85% was to generate the hypotheses and 15% for the premises.

To evaluate the quality of the results, we ran a full validation executed by humans that revised all sentences and labels. The manual validation concluded that labels were correctly assigned in almost all cases, and only 2.6% of the instances had issues with quality.

We trained a model with InferBR and used it to predict labels from other datasets in Portuguese and vice-versa. Models trained on InferBR were better at recognizing entailment in the other Portuguese datasets than the other way around. This may indicate a better generalization power.

This chapter delivered the main contribution of this thesis *C5. A synthetic dataset for NLI in Portuguese, indicating the quality of the instances after human validation*, *i.e.,* the InferBR synthetic dataset.

6 A SENTENCE-TRANSFORMER MODEL FOR PORTUGUESE

In this chapter, we address the last research question:

RQ5. How to use the NLI dataset to train a sentence embedding model for Portuguese? How do we evaluate results?

We took advantage of our synthetic NLI dataset to train an ST model. The embeddings generated by the ST model for Portuguese datasets were used in downstream tasks and the task performance was evaluated.

ST models are based on Siamese network architecture, which is well-suited for learning sentence-level representations because it can learn from sentence comparisons. SBERT (REIMERS; GUREVYCH, 2019) is a modification of the pre-trained BERT network that uses Siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine similarity. These models produce high-quality embeddings that are useful in various downstream applications, such as semantic search, text clustering, or information retrieval. These models are also successfully applied to few-shot learning (TUNSTALL et al., 2022) in text classification datasets.

We fine-tuned pre-trained ST models using data extracted from Portuguese NLI datasets. The resulting models were used to generate embeddings for Portuguese datasets, which were then employed to train task-specific models for three types of downstream tasks: *classification*, *clustering*, and *semantic textual similarity* (STS). The process is basically the same in all tasks – the embeddings are obtained for the instances in the specific datasets and used to accomplish the task.

Details about the datasets, the evaluation metrics, the training process, and the results are described in the following sections.

6.1 Evaluation metrics

To evaluate the results of the *clustering* task, we used the v-measure (ROSENBERG; HIRSCHBERG, 2007), an external entropy-based cluster evaluation metric. External evaluation measures for clustering can be applied when class labels for each data point are given, which is the case for the datasets we selected for testing. V-measure combines two desirable clustering aspects: homogeneity and completeness. A clustering result fulfills homogeneity if all clusters contain exclusively data points from a single class. It meets completeness if all the data points that are components of a given class are parts of the same cluster. We used the v-measure implementation available in `scikit-learn`[1]

[1] <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.v_measure_score.html>

The STS task uses datasets with pairs of sentences labeled with a similarity score. The goal is to check how the cosine distance between the embeddings of the two sentences correlates with a human-labeled similarity score (CONNEAU; KIELA, 2018). The evaluation of this task was done through Pearson correlation.

For the classification task, we report Accuracy and macro-F1, which are calculated as described in Section 2.3.2.

6.2 Datasets for Evaluation

For the clustering task, we used the Globo[2] dataset and only the five biggest classes from Folha[3] dataset. The former are news extracted from sites of the Globo Group (Brazilian media conglomerate) between 2014 and 2020. The Folha dataset contains news articles from Folha de São Paulo (Brazilian Newspaper) collected between January 2015 and September 2017. Details about class distributions are presented in Table 6.1.

In the classification task, we tested embedding quality on the binary classification for offensive discourse in the dataset named HateBR (VARGAS et al., 2022). Details on class distribution are also in Table 6.1. Additionally, we generated embeddings also for the City-tweets dataset. As discussed in Section 4.2.3.2, when analyzing the experiments of DA in City-tweets, we compared the classification using the SetFit framework with a multilingual ST model. Here, we used ST models fine-tuned for Portuguese with NLI resources, including our generated synthetic NLI dataset, InferBR. We can compare these results with previous DA methods. Details on the distribution of the 16 classes of the City-tweets dataset can be found in Table 4.7.

For the STS task, we used the test set of ASSIN2 (details in Table 5.3), which was not used to fine-tune the ST models. Besides the NLI labels, the dataset also contains a relatedness score ranging from 1 (completely different sentences on different topics) to 5 (sentences mean essentially the same thing).

6.3 Training Settings

This section describes the settings to train the ST models to adapt them to Portuguese, detailed in Section 6.3.1. The configuration used in the downstream tasks for employing the embeddings generated from the resulting ST models is depicted in Section 6.3.2.

[2]https://www.kaggle.com/datasets/diogocaliman/notcias-publicadas-no-brasil/data
[3]https://www.kaggle.com/datasets/marlesson/news-of-the-site-folhauol/data

Table 6.1 – Datasets used to evaluate ST models

Task	Dataset	Classes	Instances
Clustering	Globo	Economy	1,558
		Sports	6,030
		Celebrities	540
		Politics	1,363
		Technology	612
	Folha	Everyday life	16,967
		Sports	19,730
		Market	20,970
		World	17,130
		Politics	22,022
Classification	HateBR	Non-ofensive	3,500
		Ofensive	3,500
STS	ASSIN2	na	2,448

Note: The ASSIN2 refers to the test set.

6.3.1 Portuguese ST model

There are several ways to organize the data to train ST models. The NLI classes are well suited to the positive and negative pairs organization required to fine-tune those models (REIMERS; GUREVYCH, 2019). We used the synthetic dataset InferBR, and the manually created ASSIN2 as training data. We also translated 152k entailment pairs of the SNLI dataset from English to Portuguese. We refer to this dataset as **snliT**. Only training data of these datasets were used.

Two kinds of training were done: *fine-tuning existing multilingual ST models* and *training a new ST model* using an existing BERT model and its weights to configure the two Siamese networks. In the fine-tuning approach, two multilingual pre-trained models were used. The first is the paraphrase-multilingual-mpnet-base-v2 (REIMERS; GUREVYCH, 2020) named here as **M1**, which was trained using XLMRoberta architecture with a max sequence length of 128 tokens. The second is multilingual-e5-small (WANG et al., 2022a), named **M2**, trained using BERT architecture with a max length of 512. The max length sequence determines how many tokens from the text are encoded by the model. We adopted the Multiple Negative Ranking Loss (HENDERSON et al., 2017), which requires batches containing only positive pairs (the entailment pairs of our training data). We ensure that each batch does not have the same sentence occurring multiple times. The learning rate was 2e-5, batch sizes of 128, the optimizer was AdamW, and the weight decay was set to 0.01. We trained models for eight epochs and selected the model that performed better on the ASSIN2 validation set using the cosine similarity. We did not tune hyperparameters but tested different combinations of training data: (a) only InferBR and (b) InferBR and ASSIN2.

We used the BERTimbau model architecture to train a new ST model named **M3**. The model weights are used to initialize the Siamese network. We kept 512 tokens as the max length

and used mean-pooling to average the embeddings across all words in the sentence. We kept 768 dimensions to the resulting embeddings, meaning it encodes 512 tokens of an input sentence to a single vector with 768 dimensions. The hyperparameters used were the same as in the fine-tuning procedure described above in this Section.

6.3.2 Downstream tasks

For **clustering**, we used the fast method mini-batch k-means (NEWLING; FLEURET, 2016) with a batch size of 32 and the number of clusters equal to the number of classes in the dataset. We repeated clustering the embeddings 100 times for each ST model in Globo and Folha datasets, reporting the average v-measure and the standard deviation.

For **classification**, we used two approaches to train the classifier with the embeddings of the ST models. For HateBR, split holdout 25% of the dataset for testing, we encoded the remaining 75% of the training and used the resulting embeddings to train a Logistic Regression (LR) classifier with a maximum number of iterations of 200. We execute this process 25 times with different seeds for the random state. Since the City-tweets dataset is small and highly imbalanced, we used the SetFit framework to train the classifier. We want to understand the effect of the Portuguese specialization we did on the multilingual ST models and analyze these results compared to the experiments run in 4.2.3.2. In SetFit, we fine-tuned the ST models with 70% of the City-tweets training set, leaving 25% for evaluation during training. We applied the same configuration described in the final of Section 4.2.2, but besides the multilingual models, we used all Portuguese ST models trained. We kept the default classification head from SetFit, a logistic regression model with 200 iterations.

For **STS**, we calculated the semantic similarity using the cosine. Embeddings of the ASSIN2 test set were generated for each ST model, calculating the Pearson correlation between the ground truth and the calculated cosine using the embedding space of each ST model.

6.4 Results

This section presents the ST models trained with the Portuguese NLI resources, including InferBR. The results of clustering embeddings are presented in Table 6.2. When fine-tuning the pre-trained ST models M1 and M2, the resulting models yielded average v-measure slicing higher than the respective multilingual pre-trained model, except for M2.1. However, the differences are very small, considering the standard deviation. There are several opportunities for improvement since we did not optimize hyperparameters. M3, an ST model trained from scratch and initial-

Table 6.2 – Results of embeddings for clustering

Dataset	Model ID	ST Model or training strategy	v-measure	std
Globo	M1	paraphrase-multilingual-mpnet-base-v2	0.501	0.082
Globo	M1.1	fine-tuning M1 (infer+assin)	0.538	0.052
Globo	M1.2	fine-tuning M1 (infer)	0.517	0.058
Globo	M2	multilingual-e5-small	0.592	0.056
Globo	M2.1	fine-tuning M2 (infer+assin)	0.591	0.056
Globo	M2.2	fine-tuning M2 (infer)	0.597	0.060
Globo	M3	from scratch with BERTimbau weights	**0.601**	0.064
Folha	M1	paraphrase-multilingual-mpnet-base-v2	0.388	0.055
Folha	M1.1	fine-tuning M1 (infer+assin)	0.408	0.038
Folha	M1.2	fine-tuning M1 (infer)	0.402	0.053
Folha	M2	multilingual-e5-small	0.485	0.064
Folha	M2.1	fine-tuning M2 (infer+assin)	0.482	0.055
Folha	M2.2	fine-tuning M2 (infer)	0.507	0.052
Folha	M3	from scratch with BERTimbau weights	**0.531**	0.063

Note: The lines group each dataset and pre-trained model with the respective fine-tuned versions. M3 model was trained with infer+assin+snliT.

ized with BERTimbau weights, achieved the highest score for both datasets. The best-fine-tuned models in Globo and Folha yielded around 4.8% on average higher results than the multilingual models. When observing the model trained from scratch, M3, it could perform 9.4% better in Folha than the best multilingual model, while for Globo, the improvement is much lower, around 1.5%.

We can also see that multilingual model M1 is around 18% worse than M2 for clustering. We hypothesize this is related to the number of tokens each model encodes: M1 has a max length of 128 tokens, while M2 is 512. Both datasets have a high number of tokens per document. Another point contributing to this is the different tokenizers: M1, based on XLMRoberta, produces 40% more tokens than M2, based on BERT.

Table 6.3 has the classification results. The models specialized for Portuguese always yielded superior results compared to the respective multilingual pre-trained model. For HateBR, the best ST model is M3, but for City-tweets, where we used the few-shot SetFit framework, we got better macro-F1 with the fine-tuned model M1.1. Resuming the discussion started in Section 4.2.3.2 where we generally compared winning DA methods with few-shot and zero-shot learning methods for City-tweets, we see improvements in the SetFit results due to the Portuguese' models M1.1, M1.2, and M3. This shows the value of having NLI sources (part of them generated synthetically) to improve existing ST models that can then be used in a few-shot strategy like the one used by SetFit.

Analyzing the overall results for classification in terms of macro-F1, we can see that for HateBR there is an improvement of 0.8% on average over the respective pre-trained multilingual model used as the starting for the fine-tuning process. At the same time, M3 got 2.5% over the

Table 6.3 – Results for classification

Dataset	Approach	Model ID	ST Model or training strategy	Acc	F1
HateBR	emb → LR	M1	paraphrase-multilingual-mpnet-base-v2	0.865	0.865
HateBR	emb → LR	M1.1	fine-tuning M1 (infer+assin)	0.871	0.871
HateBR	emb → LR	M1.2	fine-tuning M1 (infer)	0.865	0.865
HateBR	emb → LR	M2	multilingual-e5-small	0.843	0.843
HateBR	emb → LR	M2.1	fine-tuning M1 (infer+assin)	0.854	0.854
HateBR	emb → LR	M2.1	fine-tuning M1 (infer)	0.844	0.844
HateBR	emb → LR	M3	from scratch with BERTimbau weights	**0.887**	**0.887**
City-tweets	SetFit	M1	paraphrase-multilingual-mpnet-base-v2	0.820	0.707
City-tweets	SetFit	M1.1	fine-tuning M1 (infer+assin)	0.835	**0.733**
City-tweets	SetFit	M1.2	fine-tuning M1 (infer)	0.828	0.710
City-tweets	SetFit	M2	multilingual-e5-small	0.797	0.621
City-tweets	SetFit	M2.1	fine-tuning M1 (infer+assin)	0.817	0.715
City-tweets	SetFit	M2.2	fine-tuning M1 (infer)	0.817	0.686
City-tweets	SetFit	M3	from scratch with BERTimbau weights	**0.848**	0.725

Note: The lines group each dataset and pre-trained model with the respective fine-tuned versions. M3 model was trained with infer+assin+snliT. The SetFit approach considers fine-tuning the ST model and a logistic regression head. "Emb → LR" means Embeddings directly used to train Logistic Regression. The F1 score is macro averaged.

Table 6.4 – Results for semantic textual similarity

Model ID	ST Model and fine-tuning data	Pearson
M1	paraphrase-multilingual-mpnet-base-v2	0.7836
M1.1	fine-tuning M1 (infer+assin)	**0.8113**
M1.2	fine-tuning M1 (infer)	0.8076
M2	multilingual-e5-small	0.7850
M2.1	fine-tuning M1 (infer+assin)	0.7940
M2.1	fine-tuning M1 (infer)	0.7939
M3	from scratch with BERTimbau weights	0.7912

Note: The lines group each pre-trained model and the respective fine-tuned versions. M3 model was trained with infer+assin+snliT.

best multilingual model. For City-tweets classification, the fine-tuned models have improved by 7.4% on average over the respective multilingual model they used as the starting point. M3 got 2.5% over the best multilingual model.

The STS task outcomes are presented in Table 6.4. Again, all ST models fine-tuned to Portuguese yielded better results than the respective multilingual base versions, yielding an average improvement of 2.3%. M1.1 macro-F1 is 3.35% higher than the best multilingual model. Although M3, trained from scratch with all data available, achieved higher results than both multilingual pre-trained ST models, it could not overcome any fine-tuned model. This step deserves further investigation in future works since improving this task may benefit the others in that model.

6.5 Summary and Discussion

Developing NLI resources for specific languages helps to improve ST models. Those models can produce good embeddings that improve some tasks more clearly, such as clustering, which is used today to develop good topic models, helping to discover knowledge in large amounts of text. Besides clustering, the classification of small datasets may also benefit from the ST models in few-shot approaches, as we could see for the dataset City-tweets with the SetFit framework.

Following is the answer to the last research question of this thesis:

RQ5. How to use the NLI dataset to train a sentence embedding model for Portuguese? How do we evaluate results?

We used the entailment pairs from InferBR, ASSIN2, and snliT for training ST models, which already accomplished some improvement. In future work, we will address the use of other loss functions that will also consider contradictions and neutral pairs. The results were evaluated using three downstream tasks. In the future, we may include more tasks (question answering, information retrieval, etc.) and data variation within the task (clustering other datasets than news, classification datasets with more classes, and about different domains).

In summary, the results of training ST models for Portuguese were as expected: specialization leads to better ST models, leading to better embeddings that can be used in a set of applications. However, deeper investigations are needed since the improvements are still small for some tasks and datasets. We see a vast set of options to improve the training process we did for the ST models, for example, optimizing hyperparameters and trying other loss functions, which will also add contradictions and neutral pairs to the training.

This chapter delivered the last contribution of this thesis, namely *C7. A model to generate embeddings for Portuguese sentences*.

7 CONCLUSION AND FUTURE WORK

In this work, we produced a synthetic dataset for NLI using LLM that can follow instructions in Portuguese. To achieve the goal, we first investigated the behavior of data augmentation (DA) methods in a less complex task, multiclass single-label text classification in English and Portuguese. We generated text from an initial context containing the original instance to produce an augmented synthetic version. We compared these approaches to established DA methods and also to some few-shot learning strategies. We found out that text generation as a DA method yielded good classification results.

Using GPT-3.5+ to augment our dataset in Portuguese generated unexpected high-quality sentences that were grammatically well-written, label-preserving, and very creative. We adapted the text generation approach used for DA to generate the NLI dataset with GPT-4. For 59% of the premises, we extracted a raw caption description from PraCegoVer. We employed summarization with additional cleaning instructions using GPT-4. We also innovated on generating the remaining 41% of the premises using as initial context only 16% of all unique sentences of an existing NLI dataset, SICK-BR. This approach produced very different sentences by requesting similar contexts. We developed a list of good examples that were randomly used to generate the hypotheses for each premise available, which brought good diversity. Finally, we trained ST models using the synthetic NLI dataset and concluded their embeddings were better than multilingual models for performing clustering, classification, and semantic similarity in Portuguese texts.

The purpose of DA in this thesis was to leverage the knowledge of producing synthetic data with original instances as part of the initial context used as input in LLM. We compared conventional techniques to text generation and compared their results both quantitatively and quantitatively. Our goal here was not to propose a new DA method – several ideas can be explored to improve augmentation methods for Portuguese. Various research fronts can be approached to augment data to train robust models for text classification, for example, using semi-supervised techniques as active learning, developing criteria to filter augmented instances that contribute more, removing issues from the augmented dataset using approaches such as noisy label detection, and many others.

Regarding the data sources for raw premises in Process A (see Figure 5.2a), we limited our work to an image-captioning dataset. Nevertheless, we believe the process can be easily adapted to generate premises from existing unlabeled corpora, adding more diversity in terms of genre to be more representative of the country's culture. Adapting the premise quality flags to work in other corpora and expanding the dataset in future projects will allow us to get good representations that can be used in other tasks.

This work relied on proprietary models. Yet, the OpenAI costs involved in the experiments

were relatively low, considering that the dataset InferBR has around 10k instances. The human reviews revealed high agreement with the generated label, and only 2.6% of the instances were considered low quality because the automatic label was different from the one assigned by humans or because the text was confusing due to issues in GPT-4 with common sense.

The number of annotators we used was small but specialized. However, we recognize that more reviewers are needed for the validation if we expand our dataset. If the dataset becomes too large, sampling techniques could be applied, similar to what was done in (BOWMAN et al., 2015b). The reviewers who volunteered to participate in this project could annotate labels at a reasonable pace and always had an open channel to send questions or concerns.

We did not run parameter optimization when training the ST models. There is an important space for improvement here, particularly in using other loss functions that can handle contradiction and neutral classes. We will address this in future work, including research on the best models to be used as a starting point for the Siamese network training. We also will improve the validation of the embedding quality, including more diversity in tasks and datasets.

The ethical considerations regarding this thesis are related to using LLM for text generation, especially GPT-4, which does not release information on its training data. All data we used to produce synthetic instances for the text classification experiments and any data source used to generate the NLI dataset or any dataset used as a benchmark are public, and no private data is involved. All automatically generated sentences were manually checked, and although nothing caught the attention of the reviewers, they may still contain societal biases, for example, a prevalence of instances with specific race, ethnicity, gender, age, religion, abilities, socioeconomic profile, *etc.* This may be part of future work, not depending exclusively on reviewers' judgment but also using existing models and tools to help identify and mitigate those biases.

The evolution of language models has occurred very fast, especially in the last year. While this thesis was written, several open models specialized in following human instructions were launched. However, we face infrastructure limitations and very large models are sometimes difficult to fit in affordable hardware, which was one of the reasons to go for the OpenAI API. Today, the Chatbot Maritaca[1], trained for Portuguese, also provides an API. The service is not free, but its authors mentioned prices are lower than OpenAI's. Maritaca is based on Sabiá (PIRES et al., 2023), a language model that further pre-trains LLaMA (TOUVRON et al., 2023) in Portuguese texts using 3% or less of their original pretraining budget. Sabiá outperformed the multilingual models, which struggled to generate answers in Portuguese with very low scores in native text. The best model, Sabiá-65B surpasses GPT-3.5-Turbo and Llama 65B in native and translated text, while GPT-4 still performed 16% better. Future work will test our method using Maritaca and other models that may come soon to produce the synthetic instances.

[1] https://www.maritaca.ai/

REFERENCES

AKOJU, S. A. et al. Synthetic dataset for evaluating complex compositional knowledge for natural language inference. In: **Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)**. Toronto, Canada: Association for Computational Linguistics, 2023. p. 157–168.

ALAMMAR, J. The illustrated gpt-2 (visualizing transformer language models). **http://jalammar.github.io/**, Aug 2019. Available from Internet: <http://jalammar.github.io/illustrated-gpt2/>.

AMJAD, M.; SIDOROV, G.; ZHILA, A. Data augmentation using machine translation for fake news detection in the urdu language. In: **Proceedings of The 12th Language Resources and Evaluation Conference**. [S.l.: s.n.], 2020. p. 2537–2542.

ANABY-TAVOR, A. et al. Do not have enough data? deep learning to the rescue! In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2020. v. 34, n. 05, p. 7383–7390.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. In: BENGIO, Y.; LECUN, Y. (Ed.). **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**. [s.n.], 2015. Available from Internet: <http://arxiv.org/abs/1409.0473>.

BAYER, M.; FREY, T.; REUTER, C. Multi-level fine-tuning, data augmentation, and few-shot learning for specialized cyber threat intelligence. **Computers & Security**, v. 134, 2023.

BAYER, M.; KAUFHOLD, M.-A.; REUTER, C. A survey on data augmentation for text classification. **ACM Computing Surveys**, ACM New York, NY, v. 55, n. 7, p. 1–39, 2022.

BEDDIAR, D. R.; JAHAN, M. S.; OUSSALAH, M. Data expansion using back translation and paraphrasing for hate speech detection. **Online Social Networks and Media**, Elsevier, v. 24, p. 100153, 2021.

BENCKE, L.; CECHINEL, C.; MUNOZ, R. Automated classification of social network messages into smart cities dimensions. **Future Generation Computer Systems**, v. 109, 2020.

BENCKE, L.; MOREIRA, V. P. Data augmentation strategies to improve text classification: a use case in smart cities. **Language Resources and Evaluation**, Springer, 2023. Available from Internet: <https://doi.org/10.1007/s10579-023-09685-w>.

BENGIO, Y.; DUCHARME, R.; VINCENT, P. A neural probabilistic language model. **Advances in neural information processing systems**, v. 13, 2000.

BENTHEM, J. V. **A brief history of natural logic**. 2008.

BISHOP, C. M.; NASRABADI, N. M. **Pattern recognition and machine learning**. [S.l.]: Springer, 2006.

BODY, T. et al. Using back-and-forth translation to create artificial augmented textual data for sentiment analysis models. **Expert Systems with Applications**, Elsevier, v. 178, p. 115033, 2021.

BOJANOWSKI, P. et al. Enriching Word Vectors with Subword Information. **Transactions of the Association for Computational Linguistics**, v. 5, p. 135–146, 2017. ISSN 2307-387X. Available from Internet: <https://doi.org/10.1162/tacl_a_00051>.

BOUTHILLIER, X. et al. **Dropout as data augmentation**. 2016.

BOWMAN, S. R. et al. A large annotated corpus for learning natural language inference. Association for Computational Linguistics, Lisbon, Portugal, p. 632–642, sep. 2015. Available from Internet: <https://aclanthology.org/D15-1075>.

BOWMAN, S. R. et al. A large annotated corpus for learning natural language inference. In: **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**. [S.l.]: Association for Computational Linguistics, 2015.

BROMLEY, J. et al. Signature verification using a" siamese" time delay neural network. **Advances in neural information processing systems**, v. 6, 1993.

BROWN, T. et al. Language models are few-shot learners. **Advances in neural information processing systems**, v. 33, p. 1877–1901, 2020.

CHAN, S. et al. Data distributional properties drive emergent in-context learning in transformers. In: KOYEJO, S. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2022. v. 35, p. 18878–18891. Available from Internet: <https://proceedings.neurips.cc/paper_files/paper/2022/file/77c6ccacfd9962e2307fc64680fc5ace-Paper-Conference.pdf>.

CHEN, J.; CHOI, E.; DURRETT, G. Can NLI models verify QA systems' predictions? In: MOENS, M.-F. et al. (Ed.). **Findings of the Association for Computational Linguistics: EMNLP 2021**. Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021. p. 3841–3854. Available from Internet: <https://aclanthology.org/2021.findings-emnlp.324>.

CHEN, J. et al. An empirical survey of data augmentation for limited data learning in nlp. **Transactions of the Association for Computational Linguistics**, MIT Press, v. 11, p. 191–211, 2023.

CHRISTIANO, P. F. et al. Deep reinforcement learning from human preferences. **Advances in neural information processing systems**, v. 30, 2017.

CONNEAU, A.; KIELA, D. SentEval: An evaluation toolkit for universal sentence representations. In: **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)**. Miyazaki, Japan: European Language Resources Association (ELRA), 2018. Available from Internet: <https://aclanthology.org/L18-1269>.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In: **Proceedings of NAACL-HLT**. [S.l.: s.n.], 2019. p. 4171–4186.

EDUNOV, S. et al. Understanding back-translation at scale. In: RILOFF, E. et al. (Ed.). **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**. Brussels, Belgium: Association for Computational Linguistics, 2018. p. 489–500. Available from Internet: <https://aclanthology.org/D18-1045>.

FAN, A.; LEWIS, M.; DAUPHIN, Y. Hierarchical neural story generation. In: GUREVYCH, I.; MIYAO, Y. (Ed.). **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 889–898. Available from Internet: <https://aclanthology.org/P18-1082>.

FENG, S. Y. et al. A survey of data augmentation approaches for NLP. In: **Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021**. Online: Association for Computational Linguistics, 2021. p. 968–988. Available from Internet: <https://aclanthology.org/2021.findings-acl.84>.

FENOGENOVA, A. Russian paraphrasers: Paraphrase with transformers. In: **Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing**. [S.l.: s.n.], 2021. p. 11–19.

FERREIRA, T. M.; COSTA, A. H. R. Deepbt and nlp data augmentation techniques: a new proposal and a comprehensive study. In: SPRINGER. **Brazilian Conference on Intelligent Systems**. [S.l.], 2020. p. 435–449.

FILHO, J. A. W. et al. The brWaC corpus: A new open resource for Brazilian Portuguese. In: **Intl. Conf. on Language Resources and Evaluation (LREC)**. [S.l.: s.n.], 2018.

FREITAG, M.; AL-ONAIZAN, Y. Beam search strategies for neural machine translation. In: LUONG, T. et al. (Ed.). **Proceedings of the First Workshop on Neural Machine Translation**. Vancouver: Association for Computational Linguistics, 2017. p. 56–60. Available from Internet: <https://aclanthology.org/W17-3207>.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. **Journal of the american statistical association**, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937.

GAO, T.; YAO, X.; CHEN, D. Simcse: Simple contrastive learning of sentence embeddings. In: MOENS, M.-F. et al. (Ed.). **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**. Association for Computational Linguistics, 2021. Available from Internet: <https://aclanthology.org/2021.emnlp-main.552>.

GARCEA, F. et al. Data augmentation for medical imaging: A systematic literature review. **Computers in Biology and Medicine**, Elsevier, p. 106391, 2022.

GARCÍA-PALOMARES, J. C. et al. City dynamics through twitter: Relationships between land use and spatiotemporal demographics. **Cities**, Elsevier, v. 72, p. 310–319, 2018.

GARG, S.; RAMAKRISHNAN, G. BAE: BERT-based adversarial examples for text classification. In: WEBBER, B. et al. (Ed.). **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Online: Association for Computational Linguistics, 2020. p. 6174–6181. Available from Internet: <https://aclanthology.org/2020.emnlp-main.498>.

GARG, S. et al. What can transformers learn in-context? a case study of simple function classes. In: KOYEJO, S. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2022. v. 35, p. 30583–30598. Available from Internet: <https://proceedings.neurips.cc/paper_files/paper/2022/file/c529dba08a146ea8d6cf715ae8930cbe-Paper-Conference.pdf>.

GLENN, P. et al. Jetsons at the finnlp-2023: Using synthetic data and transfer learning for multilingual esg issue classification. In: **Proceedings of the Fifth Workshop on Financial Technology and Natural Language Processing and the Second Multimodal AI For Financial Forecasting**. [S.l.: s.n.], 2023. p. 133–139.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016.

GULLI, A. AG's corpus of news articles. 2005.

HARALABOPOULOS, G. et al. Text data augmentations: Permutation, antonyms and negation. **Expert Systems with Applications**, Elsevier, v. 177, p. 114769, 2021.

HEDDERICH, M. A. et al. A survey on recent approaches for natural language processing in low-resource scenarios. In: **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**. Association for Computational Linguistics, 2021. Available from Internet: <https://aclanthology.org/2021.naacl-main.201.pdf>.

HENDERSON, M. et al. Efficient natural language response suggestion for smart reply. **arXiv preprint arXiv:1705.00652**, 2017.

HERDAĞDELEN, A. Twitter n-gram corpus with demographic metadata. **Language resources and evaluation**, Springer, v. 47, n. 4, p. 1127–1147, 2013.

HERNÁNDEZ-GARCÍA, A.; KÖNIG, P. Data augmentation instead of explicit regularization. **arXiv preprint arXiv:1806.03852**, 2018.

HINTON, G.; VINYALS, O.; DEAN, J. Distilling the knowledge in a neural network. In: **NIPS Deep Learning and Representation Learning Workshop**. [s.n.], 2015. Available from Internet: <http://arxiv.org/abs/1503.02531>.

HOLTZMAN, A. et al. The curious case of neural text degeneration. In: **International Conference on Learning Representations**. [s.n.], 2019. Available from Internet: <https://openreview.net/forum?id=rygGQyrFvH>.

ISO. **ISO 37120:2014 - Sustainable development of communities — Indicators for city services and quality of life**. [S.l.], 2014.

JURAFSKY, D.; MARTIN, J. H. **Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition**. [S.l.]: Prentice Hall, 2023.

KALOULI, A.-L. et al. Explaining simple natural language inference. In: FRIEDRICH, A.; ZEYREK, D.; HOEK, J. (Ed.). **Proceedings of the 13th Linguistic Annotation Workshop**. Florence, Italy: Association for Computational Linguistics, 2019. p. 132–143. Available from Internet: <https://aclanthology.org/W19-4016>.

KAMATH, U.; LIU, J.; WHITAKER, J. **Deep learning for NLP and speech recognition**. [S.l.]: Springer, 2019.

KATZ, J. J. **Semantic Theory**. New York,: Harper & Row, 1972.

KEEN, E. **Gartner Identifies Top Trends Shaping the Future of Data Science and Machine Learning**. 2023. Available from Internet: <https://www.gartner.com/en/newsroom/press-releases/2023-08-01-gartner-identifies-top-trends-shaping-future-of-data-science-and-machine-learning>.

KIM, H. H. et al. Alp: Data augmentation using lexicalized pcfgs for few-shot text classification. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2022. v. 36, n. 10, p. 10894–10902.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: BENGIO, Y.; LECUN, Y. (Ed.). **3rd International Conference on Learning Representations, ICLR**. [s.n.], 2015. Available from Internet: <http://arxiv.org/abs/1412.6980>.

KOBAYASHI, S. Contextual augmentation: Data augmentation by words with paradigmatic relations. In: WALKER, M.; JI, H.; STENT, A. (Ed.). **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)**. New Orleans, Louisiana: Association for Computational Linguistics, 2018. p. 452–457. Available from Internet: <https://aclanthology.org/N18-2072>.

LAI, V. et al. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In: FENG, Y.; LEFEVER, E. (Ed.). **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**. Singapore: Association for Computational Linguistics, 2023. p. 318–327. Available from Internet: <https://aclanthology.org/2023.emnlp-demo.28>.

LEWIS, M. et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: JURAFSKY, D. et al. (Ed.). **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**. Online: Association for Computational Linguistics, 2020. p. 7871–7880. Available from Internet: <https://aclanthology.org/2020.acl-main.703>.

LI, Z. et al. Synthetic data generation with large language models for text classification: Potential and limitations. In: **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**. Singapore: Association for Computational Linguistics, 2023. p. 10443–10461. Available from Internet: <https://aclanthology.org/2023.emnlp-main.647>.

LIN, T.-Y. et al. Microsoft coco: Common objects in context. In: **Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13**. [S.l.]: Springer, 2014.

LIU, A. et al. WANLI: Worker and AI collaboration for natural language inference dataset creation. In: GOLDBERG, Y.; KOZAREVA, Z.; ZHANG, Y. (Ed.). **Findings of the Association for Computational Linguistics: EMNLP 2022**. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022. p. 6826–6847. Available from Internet: <https://aclanthology.org/2022.findings-emnlp.508>.

LIU, P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 55, n. 9, jan 2023. ISSN 0360-0300. Available from Internet: <https://doi.org/10.1145/3560815>.

LOSHCHILOV, I.; HUTTER, F. Decoupled weight decay regularization. In: **7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019**. OpenReview.net, 2019. Available from Internet: <https://openreview.net/forum?id=Bkg6RiCqY7>.

MANNING, C.; SCHUTZE, H. **Foundations of statistical natural language processing**. [S.l.]: MIT press, 1999.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. **An Introduction to Information Retrieval**. Springer International Publishing, 2009. Available from Internet: <https://nlp.stanford.edu/IR-book/>.

MARELLI, M. et al. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In: **Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)**. [S.l.]: Association for Computational Linguistics, 2014.

MCNALLY, L. **Existential**. 2021. Available from Internet: <https://www.oxfordbibliographies.com/view/document/obo-9780199772810/obo-9780199772810-0070.xml>.

MENG, Y. et al. Generating training data with language models: Towards zero-shot language understanding. **Advances in Neural Information Processing Systems**, v. 35, p. 462–477, 2022.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. In: BENGIO, Y.; LECUN, Y. (Ed.). **1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings**. [s.n.], 2013. Available from Internet: <http://arxiv.org/abs/1301.3781>.

MIKOLOV, T. et al. Extensions of recurrent neural network language model. In: IEEE. **2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)**. [S.l.], 2011. p. 5528–5531.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. **Advances in neural information processing systems**, v. 26, 2013.

MINAEE, S. et al. Deep learning–based text classification: A comprehensive review. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 54, n. 3, p. 1–40, 2021.

NEWLING, J.; FLEURET, F. Nested mini-batch k-means. **Advances in neural information processing systems**, v. 29, 2016.

NIKOLENKO, S. I. **Synthetic data for deep learning**. [S.l.]: Springer, 2021.

OKUR, E.; SAHAY, S.; NACHMAN, L. Data augmentation with paraphrase generation and entity extraction for multimodal dialogue system. In: **Proceedings of the Thirteenth Language Resources and Evaluation Conference**. Marseille, France: European Language Resources Association, 2022. p. 4114–4125. Available from Internet: <https://aclanthology.org/2022.lrec-1.437>.

OpenAI. **API Policy - Why did OpenAI choose to release an API instead of open-sourcing the models?** 2023. <https://help.openai.com/en/articles/4963862-why-did-openai-choose-to-release-an-api-instead-of-open-sourcing-the-models>. Online; accessed 03 November 2023.

OPENAI. **GPT-4 Technical Report**. 2023.

OSWALD, J. V. et al. Transformers learn in-context by gradient descent. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2023. p. 35151–35174.

OUYANG, L. et al. Training language models to follow instructions with human feedback. **arXiv preprint arXiv:2203.02155**, 2022.

PALEYES, A.; URMA, R.; LAWRENCE, N. D. Challenges in deploying machine learning: a survey of case studies. **CoRR**, abs/2011.09926, 2020. Available from Internet: <https://arxiv.org/abs/2011.09926>.

PELLICER, L. F. A. O.; FERREIRA, T. M.; COSTA, A. H. R. Data augmentation techniques in natural language processing. **Applied Soft Computing**, Elsevier, v. 132, p. 109803, 2023.

PENNINGTON, J.; SOCHER, R.; MANNING, C. GloVe: Global vectors for word representation. In: **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543. Available from Internet: <https://aclanthology.org/D14-1162>.

PIRES, R. et al. Sabiá: Portuguese large language models. In: NALDI, M. C.; BIANCHI, R. A. C. (Ed.). **Intelligent Systems**. Cham: Springer Nature Switzerland, 2023. p. 226–240. ISBN 978-3-031-45392-2.

PLA, F.; HURTADO, L.-F. Spanish sentiment analysis in twitter at the tass workshop. **Language Resources and Evaluation**, Springer, v. 52, n. 2, p. 645–672, 2018.

PURI, M.; VARDE, A. S.; MELO, G. de. Commonsense based text mining on urban policy. **Language Resources and Evaluation**, Springer, p. 1–31, 2022.

PyTorch Tutorials. **The Seq2Seq Model**. 2023. <https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html#the-seq2seq-model>. Online; accessed 29 October 2023.

RADFORD, A. et al. Improving language understanding by generative pre-training. **OpenAI blog**, OpenAI, 2018.

RADFORD, A. et al. Language models are unsupervised multitask learners. **OpenAI blog**, v. 1, n. 8, p. 9, 2019.

RAFFEL, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. **J. Mach. Learn. Res.**, JMLR.org, v. 21, n. 1, 2020. ISSN 1532-4435.

REAL, L.; FONSECA, E.; OLIVEIRA, H. G. The assin 2 shared task: a quick overview. In: SPRINGER. **Computational Processing of the Portuguese Language: 14th International Conference, PROPOR 2020, Evora, Portugal, March 2–4, 2020, Proceedings 14**. [S.l.], 2020. p. 406–412.

REAL, L. et al. SICK-BR: a portuguese corpus for inference. In: SPRINGER. **Computational Processing of the Portuguese Language: 13th International Conference, PROPOR 2018, Canela, Brazil, September 24–26, 2018, Proceedings 13**. [S.l.], 2018. p. 303–312.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In: **Proceed. Conf. on Empirical Methods in NLP**. [S.l.]: ACL, 2019.

REIMERS, N.; GUREVYCH, I. Making monolingual sentence embeddings multilingual using knowledge distillation. In: **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**. Association for Computational Linguistics, 2020. Available from Internet: <https://arxiv.org/abs/2004.09813>.

ROSENBERG, A.; HIRSCHBERG, J. V-measure: A conditional entropy-based external cluster evaluation measure. In: **Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)**. [S.l.: s.n.], 2007. p. 410–420.

ROSENTHAL, S.; FARRA, N.; NAKOV, P. SemEval-2017 task 4: Sentiment analysis in Twitter. In: **Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)**. [S.l.: s.n.], 2017.

SADAT, M.; CARAGEA, C. Learning to infer from unlabeled data: A semi-supervised learning approach for robust natural language inference. In: **Findings of the Association for Computational Linguistics: EMNLP 2022**. [S.l.]: Association for Computational Linguistics, 2022.

SALVATORE, F. de S. et al. A resampling-based method to evaluate nli models. **Natural Language Engineering**, Cambridge University Press, p. 1–28, 2023.

SANTOS, G. O. dos; COLOMBINI, E. L.; AVILA, S. # pracegover: A large dataset for image captioning in portuguese. **Data**, MDPI, v. 7, n. 2, p. 13, 2022.

SENNRICH, R.; HADDOW, B.; BIRCH, A. Improving neural machine translation models with monolingual data. In: ERK, K.; SMITH, N. A. (Ed.). **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Berlin, Germany: Association for Computational Linguistics, 2016. p. 86–96. Available from Internet: <https://aclanthology.org/P16-1009>.

SHORTEN, C.; KHOSHGOFTAAR, T. M.; FURHT, B. Text data augmentation for deep learning. **Journal of big Data**, Springer, v. 8, n. 1, p. 1–34, 2021.

SOCHER, R. et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: **Proceedings of the 2013 conference on empirical methods in natural language processing**. [S.l.: s.n.], 2013. p. 1631–1642.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information processing & management**, Elsevier, v. 45, n. 4, p. 427–437, 2009.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: pretrained bert models for brazilian portuguese. In: SPRINGER. **Brazilian Conference on Intelligent Systems**. [S.l.], 2020. p. 403–417.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. **Advances in neural information processing systems**, v. 27, 2014.

TEMRAZ, M.; KEANE, M. T. Solving the class imbalance problem using a counterfactual method for data augmentation. **Machine Learning with Applications**, Elsevier, v. 9, p. 100375, 2022.

TensorFlow Tutorials. **Text classification with an RNN**. 2023. <https://www.tensorflow.org/text/tutorials/text_classification_rnn?hl=pt-br>. Online; accessed 29 October 2023.

The Conversation. **AI is closer than ever to passing the Turing test for 'intelligence'. What happens when it does?** 2023. <https://theconversation.com/ai-is-closer-than-ever-to-passing-the-turing-test-for-intelligence-what-happens-when-it-does-214721>. Online; accessed 29 October 2023.

TOUVRON, H. et al. Llama: Open and efficient foundation language models. **CoRR**, abs/2302.13971, 2023. Available from Internet: <https://doi.org/10.48550/arXiv.2302.13971>.

TUNSTALL, L. et al. Efficient few-shot learning without prompts. **arXiv preprint arXiv:2209.11055**, 2022.

TURING, A. Computing machinery and intelligence. **MInd**, v. 59, p. 433–460, 1950.

VARGAS, F. A. et al. Hatebr: a large expert annotated corpus of brazilian instagram comments for offensive language and hate speech detection. In: **Conference on Language Resources and Evaluation - LREC**. [S.l.]: European Language Resources Association - ELRA, 2022.

VASWANI, A. et al. Attention is all you need. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2017. p. 5998–6008.

WANG, L. et al. Text embeddings by weakly-supervised contrastive pre-training. **CoRR**, abs/2212.03533, 2022. Available from Internet: <https://doi.org/10.48550/arXiv.2212.03533>.

WANG, S. et al. Paratag: A dataset of paraphrase tagging for fine-grained labels, nlg evaluation, and data augmentation. In: **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2022. p. 7111–7122.

WANG, X. et al. Improving natural language inference using external knowledge in the science questions domain. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2019. v. 33, n. 01, p. 7208–7215.

WEI, J. et al. Chain-of-thought prompting elicits reasoning in large language models. **Advances in Neural Information Processing Systems**, v. 35, p. 24824–24837, 2022.

WEI, J.; ZOU, K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**. Hong Kong, China: Association for Computational Linguistics, 2019. p. 6383–6389. Available from Internet: <https://www.aclweb.org/anthology/D19-1670>.

WEI, J. W. et al. Larger language models do in-context learning differently. **CoRR**, abs/2303.03846, 2023. Available from Internet: <https://doi.org/10.48550/arXiv.2303.03846>.

WEIZENBAUM, J. Eliza—a computer program for the study of natural language communication between man and machine. **Communications of the ACM**, ACM New York, NY, USA, v. 9, n. 1, p. 36–45, 1966.

WILLIAMS, A.; NANGIA, N.; BOWMAN, S. A broad-coverage challenge corpus for sentence understanding through inference. 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: Association for Computational Linguistics, 2018.

WITTEVEEN, S.; ANDREWS, M. Paraphrasing with large language models. In: BIRCH, A. et al. (Ed.). **Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP 2019, Hong Kong, November 4, 2019**. Association for Computational Linguistics, 2019. p. 215–220. Available from Internet: <https://doi.org/10.18653/v1/D19-5623>.

WU, X. et al. Conditional bert contextual augmentation. In: SPRINGER. **International Conference on Computational Science**. [S.l.], 2019. p. 84–95.

XIE, Q. et al. Unsupervised data augmentation for consistency training. In: LAROCHELLE, H. et al. (Ed.). **Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual**. [s.n.], 2020. Available from Internet: <https://proceedings.neurips.cc/paper/2020/hash/44feb0096faa8326192570788b38c1d1-Abstract.html>.

YE, J. et al. ProGen: Progressive zero-shot dataset generation via in-context feedback. In: **Findings of the Association for Computational Linguistics: EMNLP 2022**. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022. p. 3671–3683. Available from Internet: <https://aclanthology.org/2022.findings-emnlp.269>.

YOO, K. M. et al. GPT3Mix: Leveraging large-scale language models for text augmentation. In: **Findings of the Association for Computational Linguistics: EMNLP 2021**. [S.l.]: Association for Computational Linguistics, 2021.

YOUNG, P. et al. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. **Transactions of the Association for Computational Linguistics**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 2, p. 67–78, 2014.

YOUNG, P. et al. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. **Transactions of the Association for Computational Linguistics**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 2, p. 67–78, 2014.

ZHANG, H. et al. A survey of controllable text generation using transformer-based pre-trained language models. **ACM Computing Surveys**, ACM New York, NY, 2022.

ZHANG, J. et al. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2020. p. 11328–11339.

ZHANG, X.; ZHAO, J.; LECUN, Y. Character-level convolutional networks for text classification. **Advances in neural information processing systems**, v. 28, p. 649–657, 2015.

ZHANG, Y.; BALDRIDGE, J.; HE, L. PAWS: Paraphrase Adversaries from Word Scrambling. In: **Proc. of NAACL**. [S.l.: s.n.], 2019.

ZHANG, Y.; ZHOU, K.; LIU, Z. What makes good examples for visual in-context learning? **CoRR**, abs/2301.13670, 2023. Available from Internet: <https://doi.org/10.48550/arXiv.2301.13670>.

ZHAO, D. et al. Equivalence between dropout and data augmentation: A mathematical check. **Neural Networks**, v. 115, p. 82–89, 2019. ISSN 0893-6080. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0893608019300942>.

ZHAO, T. et al. Coupled social media content representation for predicting individual socioeconomic status. **Expert Systems with Applications**, Elsevier, v. 198, p. 116744, 2022.

ZHU, Z. et al. Causal language model aided sequential decoding with natural redundancy. **IEEE Transactions on Communications**, IEEE, 2023.

APPENDIX A — RESUMO EXPANDIDO

A tarefa de Inferência de Linguagem Natural (NLI) é uma subclasse da classificação de textos focada na dedução – um modelo é apresentado a um par de sentenças (premissa e hipótese) e classifica a relação entre os seus significados. Pode ser vista como uma tarefa de classificação focada na dedução (SALVATORE et al., 2023) - um modelo é apresentado com um par de sentenças e classifica a relação entre seus significados (JURAFSKY; MARTIN, 2023). A primeira frase é conhecida como *premissa* (P), e a segunda é a *hipótese* (H). Um modelo NLI deve inferir se (i) H implica P (ou seja, com base em P, podemos inferir que H é verdadeiro), (ii) H contradiz P (*i.e.,* com base em P, podemos inferir que H é falso), ou (iii) H é neutro em relação a P (a verdade de H não pode ser determinada com base em P).

Treinar modelos com conjuntos de dados para NLI é fundamental para sistemas semânticos. Além disso, conjuntos de dados NLI são usados para treinar modelos de *sentence-transformers* (ST), que usam redes Siamesas para aprender a relação entre o par de sentenças, gerando boas representações (*embeddings*) em um espaço onde sentenças semelhantes estão próximas e as diferentes estão separadas. As *embeddings* de frases podem ser usadas como recursos para treinar outros modelos em tarefas como *clustering*, por exemplo.

Os recursos existentes para NLI em português são limitados. Criar ou ampliar conjuntos de dados manualmente é custoso e requer conhecimento especializado. O aumento de dados (DA) oferece alternativas para superar esse caminho, visto que DA é o primeiro passo para o desenvolvimento de instâncias sintéticas, e a geração de texto pode ser usada como um método de DA, especialmente ao utilizar o poder dos recentes grandes modelos de linguagens (LLM).

Este trabalho se concentra na produção de um conjunto sintético de dados para NLI e na sua utilização para treinar modelos ST para gerar embeddings em português, empregando DA para classificação de texto como primeiro passo para avaliar o comportamento da geração de texto.

Nossa hipótese- é que *gerar texto usando LLM generativo recente pode ter sucesso como um método de aumento para português apresentando potencial e flexibilidade que nos permite atingir nosso objetivo principal: desenvolver um conjunto de dados NLI sintético para português e usá-lo para treinar um modelo ST para esse idioma.*

Primeiro, investigamos estratégias de aumento para classificação de texto usando métodos e técnicas de DA estabelecidas baseadas na geração de texto (TG). Montamos cenários com poucos dados em inglês e avaliamos os resultados da classificação nos conjuntos de dados aumentados. Adaptamos alguns dos métodos de DA utilizados em inglês para português, incluindo geração de texto (TG), e os aplicamos a um caso de uso. Avaliamos quantitativa e qualitativamente os resultados da classificação e as instâncias sintéticas no conjunto de dados aumentado português. Os resultados demonstram que a geração de textos no contexto de DA para classificação com grandes

modelos como GPT-3.5 e GPT-4 é competitiva e gera instâncias de alta qualidade em portuguese. Ajustamos o método de TG utilizado em DA para gerar premissas e hipóteses sintéticas para NLI em português e produzimos o dataset InferBR. Este dataset foi totalmente revisado por humanos que concordaram com as classes atribuídas aos pares e também avaliaram ser de ótima qualidade o texto gerado. Utilizamos os pares (premissa e hipótese) de InferBR para treinar um modelo de Setence-Transformers (ST). A qualidade das *embeddings* gerados por este modelo foi avaliada em conjuntos de dados em portugues para *clustering*, similaridade semântica e classificação. Os resultados demonstraram que modelos ST especializados em gerar embeddings em português, apresentaram melhor desempenho que os modelos multilíngues existentes nas tarefas de *clustering*, classificação e similaridade semântica.

As principais contribuições desta tese são:

C1. Uma análise do impacto dos métodos DA sobre cenários simulados de poucos dados usando conjuntos de dados de classificação de texto públicos bem conhecidos em inglês;

C2. Um conjunto de dados denominado City-tweets com tweets de 1993 em português rotulados com a dimensão Smart City a que se referem;

C3. A adaptação de técnicas de AD que foram originalmente concebidas para tarefas de classificação em inglês para trabalhar com o português;

C4. Uma comparação das técnicas de AD num problema de classificação multiclasse de rótulo único com o conjunto de dados português, considerando também uma análise qualitativa dos dados aumentados;

C5. Um conjunto de dados sintético para NLI em português, InferBR, indicando a qualidade das instâncias após validação humana;

C6. Os processos para geração de conjuntos de dados NLI que também podem ser adaptados para outras tarefas além do NLI e serem reutilizados por pesquisadores que trabalham em linguagens de poucos recursos; e

C7. Um modelo para gerar embeddings para sentenças em português.