UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

EDUARDO BRAGA RHODEN

Exploring Active Sampling Strategies for Self-Training in Named Entity Recognition

Work presented in partial fulfillment of the requirements for the degree of Bachelor in Computer Science

Advisor: Prof. Dr. Dennis Giovani Balreira
Co-advisor: Rafael Oleques Nunes

Porto Alegre
January 2025

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitora: Prof^{a}. Marcia Cristina Bernardes Barbosa
Vice-Reitor: Prof. Pedro de Almeida Costa
Pró-Reitora de Graduação: Prof^{a}. Nádya Pesce da Silveira
Diretor do Instituto de Informática: Prof. Luciano Paschoal Gaspary
Coordenador do Curso de Ciência de Computação: Prof. Marcelo Walter
Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

AGRADECIMENTOS

Agradeço profundamente ao Prof. Dennis e ao Rafael pela orientação e por compreenderem as dificuldades que enfrentei. Elaborar este trabalho foi um grande desafio, especialmente considerando os obstáculos de aprender uma nova área do conhecimento. No entanto, pude contar com o apoio deles em cada etapa dessa jornada.

Por fim, sou imensamente grato à minha família e aos meus amigos, que estiveram ao meu lado durante toda essa trajetória.

ABSTRACT

Named Entity Recognition (NER) is an essential task in Natural Language Processing (NLP) that focuses on detecting and categorizing named entities within text. Supervised NER models typically rely on large amounts of labeled data, which can be both costly and time-intensive to obtain. Active sampling, a technique that selects the most informative instances for labeling, has demonstrated its ability to lower labeling costs by prioritizing the most valuable data. This study examines various sampling strategies based on the BM25 (Best Match 25) algorithm within a self-training framework to fine-tune a BERT model for NER. These strategies involve selecting the most relevant sentences from an unlabeled corpus, where, for each category in the labeled dataset, the terms from all associated sentences serve as the BM25 query. The strategies vary in how they incorporate the distribution of categories within the labeled dataset. Using a Brazilian Portuguese NER dataset from the legislative domain, we assess the effectiveness of these strategies by comparing their performance against a random-based query baseline and by experimenting with different sampling fetch sizes. Although the novel sampling strategies perform on par with the baseline, they provide significant insights into the role of BM25 as a sampling method in a self-training context. These findings emphasize key challenges and identify potential directions for future research, particularly regarding the quantity and diversity of samples chosen during training iterations.

Keywords: NER. Named Entity Recogntion. Active sampling. Self-training.

Explorando Estratégias de Amostragem Ativa para Auto-Treinamento no Reconhecimento de Entidades Nomeadas

RESUMO

O Reconhecimento de Entidades Nomeadas (Reconhecimento de Entidades Nomeadas - REN) é uma tarefa essencial em Processamento de Linguagem Natural (PLN), que consiste em detectar e categorizar entidades nomeadas em textos. Modelos de NER supervisionados geralmente dependem de grandes quantidades de dados rotulados, cuja obtenção pode ser tanto custosa quanto demorada. A amostragem ativa, uma técnica que seleciona as instâncias mais informativas para rotulação, tem demonstrado sua capacidade de reduzir os custos de rotulação ao priorizar os dados mais valiosos. Este estudo analisa várias estratégias de amostragem baseadas no algoritmo BM25 (Best Match 25) em uma estrutura de auto-treinamento para fazer o *fine tuning* de um modelo BERT para NER. Essas estratégias envolvem a seleção das sentenças mais relevantes de um corpus não-rotulado, onde, para cada categoria do conjunto de dados rotulados, os termos de todas as sentenças associadas servem como consulta no BM25. As estratégias propostas diferem na forma como consideram a distribuição das categorias no conjunto de dados rotulados. Utilizando um conjunto de dados de REN em português brasileiro do domínio legislativo, avaliamos a eficácia dessas estratégias comparando seu desempenho com uma estratégia de amostragem aleatória e variando diferentes tamanhos de amostragem. Embora as novas estratégias de amostragem tenham desempenho similar à amostragem aleatória, elas fornecem entendimentos significativos sobre o papel do BM25 como método de amostragem em um contexto de auto-treinamento. Esses resultados destacam desafios importantes e identificam possíveis direções para futuras pesquisas, especialmente no que diz respeito à quantidade e diversidade das amostras escolhidas durante as iterações de treinamento.

Palavras-chave: REN. Reconhecimento de Entidade Nomeada. Amostragem ativa. Auto-treinamento.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

BCoD	Brazilian Chamber of Deputies
BERT	Bidirectional Encoder Representations from Transformers
BM25	Best Match 25
IR	Information Retrieval
NER	Named Entity Recognition
NLP	Natural Language Processing
TF-IDF	Term Frequency - Inverse Document Frequency

CONTENTS

1 INTRODUCTION

Named Entity Recognition (NER), one of the components of Natural Language Processing (NLP), focuses on identifying named entities in texts and categorizing them into predefined groups such as people, organizations, events, and others. This task can be valuable for a variety of tasks, such as enhancing search engines and Information Retrieval (IR) systems (MANNING; RAGHAVAN; SCHÜTZE, 2008), enabling other NLP tasks such as question answering (MOLLÁ; ZAANEN; SMITH, 2006), among others.

Training accurate NER models requires many annotated examples, which can be costly. To address this challenge, semi-supervised learning emerges as a promising approach, leveraging a small annotated dataset to guide the model's learning of useful patterns from a larger corpus of unlabeled data.

Naturally, the goal is to achieve a good model performance using as few resources and as little time as possible. In a semi-supervised learning context, this would mean applying active learning to select and label the most informative examples so the model can converge to good performance as quickly as possible. While it is common for the selected examples to be annotated by a human expert, also known as an oracle, the pipeline explored in this work (Figure 4.1) uses automatic annotation from a model's predictions, also known as pseudo-labeling.

This study investigates investigates IR techniques, such as BM25, to create sampling strategies with the goal of selecting the most informative samples from a large unlabeled dataset.

Although the proposed sampling strategies are domain and language-independent, the scenario explored in this work takes into account only Brazilian Portuguese legal texts.

The contributions of this work are as follows: analyzing whether the three novelty strategies could be a useful heuristic for guiding sampling in a NER self-training pipeline.

The remainder of this work is organized as follows: Chapter 2 provides the necessary background relevant to this study. Chapter 3 explores related work in the context of NER, focusing on previous research in legal-domain NER, Portuguese-language NER, and semi-supervised learning techniques for NER. Chapter 4 outlines the theoretical framework that served as a basis for the experiments. Chapter 5 details the implementation of such experiments and the results. Finally, Chapter 6 concludes the work, summarizing the key findings, discussing their implications, and suggesting directions for future research.

2 BACKGROUND

This chapter presents key concepts that are essential for understanding the proposed methodology. Section 2.1 provides an overview of NLP as described in Jurafsky and Martin (2008) and introduces NER and the stemming preprocessing technique. Section 2.2 the BERT (DEVLIN et al., 2019) language representation model is discussed, highlighting its significance in NLP research. Section 2.3 introduces two semi-supervised learning techniques, based on Settles (2012), and explains why self-training was selected over active learning for this study. Section 2.4 covers the fundamentals of IR, including the TF-IDF and BM25 algorithms, as outlined by Manning, Raghavan and Schütze (2008) and justifies the preference for BM25 in this work. Finally, section 2.5 explains different model evaluations, as described in Géron (2019), and explains the rationale behind choosing the F1-score as the evaluation metric for this research.

2.1 Natural Language Processing

Natural Language Processing (NLP) is a dynamic field that merges computer science, artificial intelligence, and linguistics to enable machines to understand and generate human language. This enablement process, however, must begin with extracting meaningful information from text, which is why Jurafsky and Martin (2008) define NLP as follows: “NLP is the field of study that focuses on the interaction between computers and human language, and, in particular, concerns itself with programming computers to fruitfully process large natural language datasets.” (2009, p. 1).

By addressing a wide range of tasks - from preprocessing techniques like stemming to fundamental tasks like tokenization and text segmentation to even more complex tasks such as machine translation and sentiment analysis — NLP aims to facilitate the interaction between humans and machines through language.

2.1.1 Named Entity Recogntion

Named Entity Recognition (NER) is a fundamental task in NLP that focuses on identifying and classifying named entities in text into predefined categories, such as persons, organizations, locations, dates, and other entities like monetary values or percentages. NER aims to enable machines to automatically recognize and categorize real-world objects or concepts mentioned in the text. This task plays a pivotal role in extracting valuable information from large volumes of unstructured text, which is common in many real-world applications such as news articles, research papers, social media posts, and customer feedback.

NER is crucial to automatically identify and retrieve information from large bodies of text, part of an area called information extraction. By recognizing entities such as names, dates, and locations, NER enables the extraction of key facts from news articles, scientific papers, and other sources of information. For example, in news monitoring, NER can identify references to people and organizations in order to track the occurrence of key events.

Furthermore, search engines and recommendation systems rely on NER to enhance their ability to understand queries and provide relevant results. For example, when users search for "Apple stock price," a NER system can correctly identify "Apple" as a company and direct the search engine to return financial data related to the company, rather than results related to the fruit.

In summary, NER is usually a building block for more complex tasks and, as it is usually at the beginning of a more complicated set of sequential tasks, its reliability greatly impacts the whole result of the set of sequential tasks.

Foundational modern NLP literature, such as Jurafsky and Martin (2008) states that modern NER techniques no longer rely solely on predefined linguistic rules, dictionaries, and pattern-matching techniques to identify and classify named entities within text because deep learning techniques have revolutionized NER.

Modern systems employ neural network architectures such as Recurrent Neural Networks (RNNs) (RUMELHART; HINTON; WILLIAMS, 1986), Long Short-Term Memory (LSTM) networks (HOCHREITER; SCHMIDHUBER, 1997), and Transformers (VASWANI et al., 2017), which include models like BERT (Bidirectional Encoder Representations from Transformers) (DEVLIN et al., 2019). These architectures leverage large pretrained language models and fine-tuning on labeled datasets, enabling them to capture nuanced semantic and syntactic relationships within text.

2.1.2 Stemming

One of the basic preprocessing techniques, stemming, refers to reducing a word to its base or root form, typically stripping off suffixes or prefixes. The goal is to remove variations of a word (such as tense, plurality, or derivational forms) so that they can be treated as the same word in subsequent processing. For example, the words "trabalhando", "trabalhador", and "trabalhou" might all be reduced to the stem "trabalh".

Stemming is particularly useful in tasks like information retrieval, where the goal is to index documents based on their semantic content rather than on the exact forms of words. However, it can sometimes produce non-standard forms that may be harder to interpret.

2.2 BERT and language-specific BERT-like models

BERT (Bidirectional Encoder Representations from Transformers) (DEVLIN et al., 2019) is a transformer-based model that has revolutionized the field of NLP. Unlike traditional models that process text in a unidirectional manner (left-to-right or right-to-left), BERT uses a bidirectional approach to learn contextualized representations of words. This innovation allows BERT to capture the full context of a word by considering both the words before and after it, which significantly improves its performance across a wide range of NLP tasks.

This model is pretrained on extensive text data using an unsupervised approach, learning to predict missing words within a sentence and identify relationships between sentence pairs. This pretraining equips BERT with a deep understanding of language structure, grammar, and context, making it highly adaptable to various downstream tasks, including text classification, named entity recognition (NER), and question answering. Moreover, BERT employs transfer learning by fine-tuning its pretrained models on specific tasks, which is particularly beneficial when annotated data is scarce. Despite being introduced in 2018, BERT and many BERT-like models continue to deliver state-of-the-art performance in many tasks, such as NER, even with the advent of large language models (NUNES et al., 2025)

The transfer learning technique allows BERT combines the power of pretrained models with task-specific fine-tuning and has significantly enhanced the accuracy and robustness of NER systems, enabling them to perform at state-of-the-art levels across diverse text corpora.

Several new BERT-like models have emerged in the last few years for different languages, like Spanish (CAÑETE et al., 2023), Arabic (ANTOUN; BALY; HAJJ, 2020) and Portuguese languages (SOUZA; NOGUEIRA; LOTUFO, 2020). All these models retain the core transformer-based architecture of BERT but they are pretrained on different language data to ensure better performance for NLP tasks in that language the model was pretrained in.

2.3 Semi-supervised Learning

While a traditional supervised learning approach works for NER, the scarcity of human-labeled data, considered ground truths in most NER scenarios, is detrimental to the accuracy of the predictions on unseen data. This scarcity could be caused by different factors, such as: high annotation cost, lack of annotators that are fluent in the language and domain specificity (e.g medical, financial, legal texts, for example). To address the scarcity problem, semi-supervised learning is typically employed.

Semi-supervised learning is a machine learning approach that addresses this scarcity issue by training a model using a small amount of labeled data and a large amount of unlabeled data. The idea is to use the unlabeled data to assist in improving the learning process, often leveraging the structure or patterns in the data. The differences between supervised and semi-supervised learning are summarized in Table 2.1

Table 2.1 – Differences between supervised and semi-supervised learning

Aspect	**Supervised learning**	**Semi-supervised learning**
Data requirements	Requires a large amount of labeled data.	Uses both a small amount of labeled data and a large amount of unlabeled data.
Goal	Learn a function that maps inputs to outputs (prediction or classification).	Use the unlabeled data to improve performance on the task, leveraging both labeled and unlabeled data.
Training Process	Model is trained on labeled data to predict labels for unseen data.	Model is trained on a small labeled dataset and refines its learning using unlabeled data.
Data availability	Requires large labeled datasets.	Requires both small labeled datasets and large unlabeled datasets

2.3.1 Active Learning

Active learning is a machine learning paradigm where a model, during its training process, actively selects the most informative or uncertain data points from a large pool of unlabeled data. These uncertain points are then presented to an oracle (usually a human annotator) to label. The model can then be retrained using both the previously labeled data and the newly labeled instances.

There are multiple active learning types, including uncertainty sampling (the model selects instances about which it has the least confidence to be labeled by a human expert), query by committee (multiple models are trained, and the instances where the models disagree are labeled by an human expert) and expected model change (selects instances that would cause the most significant change in the model if labeled by an human expert).

2.3.2 Self-training

Unfortunately, a reliable human expert is not always available. To address this issue, automatically generating labels for unlabeled data, also known as self-training, might be a possible solution.

Self-training is a semi-supervised learning technique in machine learning where a model is initially trained on a small set of labeled data. Then, the trained model is used to make predictions on a larger set of unlabeled data. The most confident predictions from the unlabeled data are added to the labeled set as pseudo-labels, and the process is repeated iteratively to improve the model's performance.

2.4 Information Retrieval

Information Retrieval (IR) refers to the process of finding relevant information from large collections of data, typically stored in databases, document collections, or the web. It involves searching for documents, records, or other types of data that match user queries, and ranking them based on relevance.

In this context, it is important to highlight two concepts: document, which is any piece of information that the system is trying to retrieve, and query, which is the input from the user and represents the information the user is seeking. After retrieving a set of documents, they are ranked based on relevance to the query.

2.4.1 Term Frequency - Inverse Document Frequency

Term Frequency - Inverse Document Frequency (TF-IDF) is a ranking function used to assess how important a word is to a document within a corpus. The goal of TF-IDF is to reflect the importance of a term within a specific document while considering its frequency in the entire corpus.

The Term Frequency (TF) part of the formula measures how frequently a word occurs in a document. The intuition is that the more often a word appears in a document, the more important it is for that document. The TF of a term t in a document d is defined as Equation 2.1:

$$\text{TF}(t, d) = \frac{\text{Number of occurrences of } t \text{ in } d}{\text{Total number of terms in } d} \tag{2.1}$$

The Inverse Document Frequency (IDF) part of the formula measures how important a word is across the entire corpus. The idea is that common words (like “the,” “is,” etc.) are not as informative, so we want to give them less weight. On the other hand, rare terms that appear in fewer documents are considered more informative and thus should have higher weight. The Inverse Document Frequency (IDF) of a term t is defined as Equation 2.2:

$$\text{IDF}(t) = \log\left(\frac{N}{\text{DF}(t)}\right) \tag{2.2}$$

Where:

- N is the total number of documents in the corpus.
- DF(t) is the number of documents in which the term t appears.

Finally, the TF-IDF score of a term t in a document d is the product of its TF and IDF values. This score indicates the importance of a word in a particular document, adjusted for its frequency in the entire corpus. And it is defined as Equation 2.3:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \tag{2.3}$$

Or, expanded, as shown in Equation 2.4:

$$\text{TF-IDF}(t,d) = \frac{\text{Number of occurrences of } t \text{ in } d}{\text{Total number of terms in } d} \times \log\left(\frac{N}{\text{DF}(t)}\right) \tag{2.4}$$

This score helps the system rank documents based on how important a term is to each document, adjusted by how rare or common the term is in the overall corpus.

2.4.2 Best Match 25

Best Match 25 (BM25) is also a ranking function. It builds on the TF-IDF approach but refines it to better handle term frequency saturation and document length variability. The BM25 score for a term t in a document d with respect to a query $Q = \{t_1, t_2, \ldots, t_k\}$ is defined as Equation 2.5:

$$\text{BM25}(d,Q) = \sum_{i=1}^{k} \text{IDF}(t_i) \times \frac{\text{TF}(t_i,d) \times (k_1+1)}{\text{TF}(t_i,d) + k_1\left(1 - b + b \times \frac{|d|}{\text{avgDL}}\right)} \tag{2.5}$$

Where:

- k_1 and b are free parameters, typically $k_1 \in [1.2, 2.0]$ and $b = 0.75$.
- $\text{TF}(t_i, d)$ is the term frequency of t_i in document d.
- $|d|$ is the length of document d (e.g., the number of terms in the document).
- avgDL is the average document length in the corpus.

One of the drawbacks with TF-IDF is that the importance of a term grows linearly with its frequency in the document. This can lead to overemphasis on very frequent terms. BM25, however, has a saturation effect. As the term frequency increases, the additional weight given to a term becomes smaller. This is controlled by the k_1 parameter.

Also, BM25 introduces length normalization to account for the fact that longer documents are more likely to contain a term multiple times simply due to their length. The b parameter controls how much length normalization is applied. This is specially useful for large documents, where term frequency in a long document could be disproportionately high compared to shorter documents.

In summary, BM25 introduces term frequency saturation and document length normalization. For this reason, BM25 was favored in this research.

2.5 Model Evaluation

In the context of Machine Learning (ML), there are multiple metrics derived from confusion matrices in a classification task, each providing a different insight of the model's performance.

For NER, it is desired to evaluate how well the model performs in terms of correctly identifying and classifying named entities (such as people, organizations, locations, dates, etc.) within text and for that end, different metrics could be used.

With the notation used in Table 2.2, which shows an example of a confusion matrix for a binary classification task, we can define such metrics.

Table 2.2 – Confusion matrix

	Predicted True	**Predicted False**
Actual True	True Positive (TP)	False Negative (FN)
Actual False	False Positive (FP)	True Negative (TN)

2.5.1 Accuracy

Accuracy is the ratio of all correctly classified instances, including both positive and negative classifications, to the total number of instances. For NER, a more refined definition of accuracy would be the proportion of correctly predicted entities (both the correct label and correct boundary) over the total number of entities (both predicted and ground truth entities).

The general form of accuracy is defined as Equation 2.6:

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{total classifications}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

Unfortunately, in cases where the dataset is imbalanced, which is very common for NER tasks, or where one type of error (FN or FP) is more costly than the other, it is better to optimize for one of the other metrics instead.

2.5.2 Precision

Precision is the ratio of the model's correctly identified positive classifications to the total number of instances it classified as positive. For NER, a more refined definition of precision would be the proportion of correctly identified entities out of all entities that the model predicted.

The general form of precision is defined as Equation 2.7:

$$\text{Precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positive}} = \frac{TP}{TP + FP} \tag{2.7}$$

It is usually the metrics used when it's crucial for positive predictions to be accurate.

2.5.3 Recall

Recall is the proportion of all actual positives that were classified correctly as positives. For NER, a more refined definition of recall would be the proportion of correctly identified entities out of all the entities that actually exist in the text.

The general form of recall is defined as Equation 2.8:

$$\text{Recall} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN} \tag{2.8}$$

It can be good metric to consider when false negatives are more expensive than false positives, therefore we would like to maximize the recall.

2.5.4 F1 score

Unfortunately, you can't have both precision and recall high. If you increase precision, it will reduce recall, and vice versa. To address this tradeoff, the F1 score, which is the harmonic mean of precision and recall, provides a single score that balances both metrics.

This metric is especially useful if the dataset contains both types of errors and for imbalanced datasets.

It is defined as Equation 2.9:

$$\mathrm{F1} = 2 * \frac{\text{precision * recall}}{\text{precision + recall}} = \frac{\text{2TP}}{\text{2TP + FP + FN}} \tag{2.9}$$

3 RELATED WORK

Even though the work proposed here in language and domain-agnostic, the experiments use data from a specific domain (legal) and a specific language (Portuguese). Both language and domain have been topic of NER research.

Since modern transformer based architectures enable transfer learning, improving NLP tasks for specific domains has been a popular topic. NER for domains like financial (SHAH et al., 2024) or medical (Fraile Navarro et al., 2023) have attracted great interest of researchers in the last few years. That is also true for the legal domain. This can be attested by the sheer number of articles about NER in the legal domain for various languages, like English (DOZIER et al., 2010), German (DARJI; MITROVIĆ; GRANITZER, 2023), Greek (ANGELIDIS; CHALKIDIS; KOUBARAKIS, 2018), Chinese (YUAN; ZHANG, 2021) among others. The fine-tuning of the BERT model using legal text in English achieved great results for NER and other challenging NLP end-tasks. (CHALKIDIS et al., 2020).

Other than domain, another interesting variable to explore in NER is the language the text is written in, regardless of its domain. A survey about the state of NER for the Portuguese language has been published (ALBUQUERQUE et al., 2023) and highlights the growth of interest in NER for Portuguese text in the last few years and cautions that the amount of research is still small when compared to other languages such as English.

Regarding semi-supervised learning, which might yield good results when there is vast amount of unlabeled data (LI; HOU; CHE, 2022), it is generally explored through active learning or self-training.

A deep learning approach for active learning for NER nearly match state-of-the-art performance with just 25% of the original training data in an active learning framework (SHEN et al., 2018). Not only that, but several articles for NER have reported better performance when using an active learning approach while not using deep learning (TRAN et al., 2017; DUPRE et al., 2020; CHEN et al., 2015).

Finally, a self-training pipeline has been proposed (NUNES et al., 2024a), using Portuguese legislative text from the UlyssesNER-BR corpus (ALBUQUERQUE et al., 2022) as a case study, achieving overall average F1-score of 86.70 ± 2.28 around the cross-validation and a final result of 90% using the BERTimbau model. Nunes' article was highly influential in the work proposed here, since this work is a generalization the self-training pipeline proposed in his article.

Although the topics explored in this work are not novel in isolation, there has been no prior research on leveraging IR techniques, such as BM25, to enhance the performance of a semi-supervised learning framework for NER. This work contributes to bridging this gap, offering new insights and opening avenues for further exploration in this area.

4 METHODOLOGY

This chapter encompasses the theoretical framework which served as a basis for the experiments that were implemented in this work. Section 4.1 explains the model evaluation strategy and metric used in this study, section 4.2 details how the self-training pipeline works and section 4.3 outlines the active sampling strategies that were implemented.

4.1 Model Evaluation

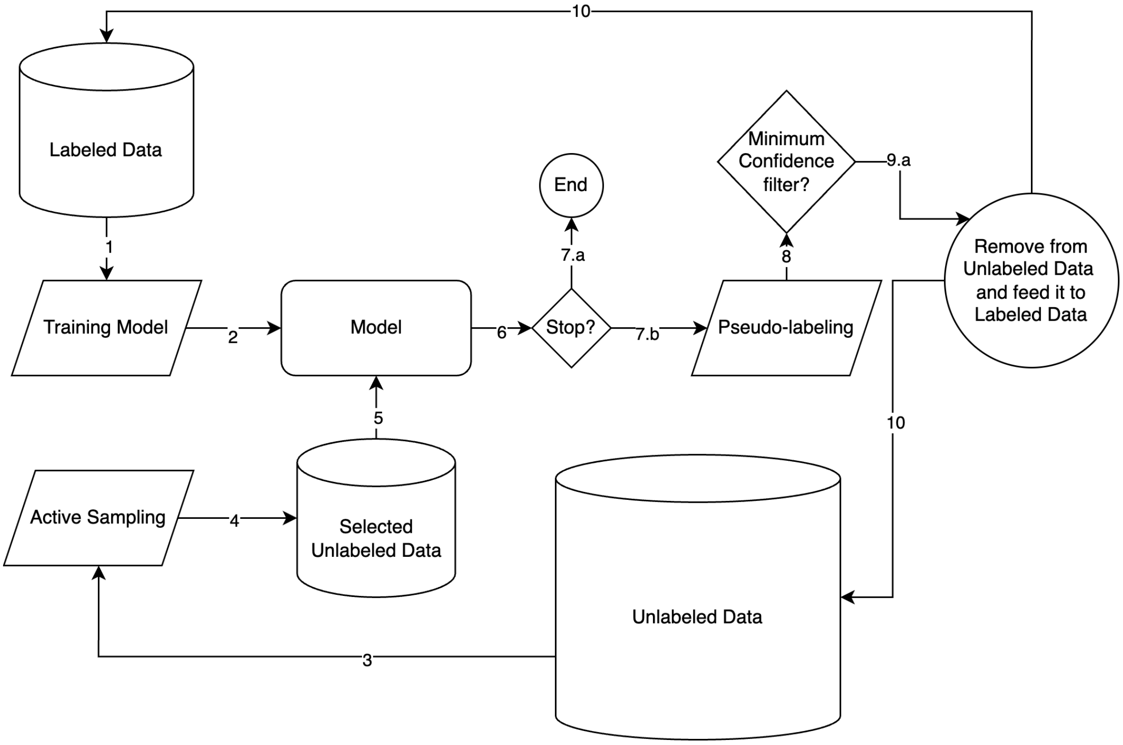
In this study, micro-averaged F1-score was chosen as the evaluation metric because it effectively balances precision and recall across all entity categories, regardless of their individual frequencies. By aggregating the contributions of true positives, false positives, and false negatives from all categories before calculating precision and recall, the micro-averaged F1-score ensures that each prediction is treated equally, which is particularly beneficial in datasets with skewed category distributions. Since NER tasks often involve highly imbalanced datasets where certain entities dominate the distribution, using micro-averaging helps avoid overemphasizing less common entities. This metric is particularly useful when the overall model's performance across all entities is of greater importance than its performance on individual categories and has been used in several papers regarding NER (ALBUQUERQUE et al., 2022; NATH; LEE; LEE, 2022).

To do so, 5-fold cross-validation was employed as the evaluation strategy to ensure robust and reliable assessment of the model's performance. This approach divides the dataset into five equally sized folds, where each fold is used as the test set exactly once, while the remaining folds are further divided into a training set and a validation set within each iteration. The training set is used to train the model, the validation set is utilized for hyperparameter tuning and early stopping, and the test set provides an unbiased estimate of the model's performance. This process was repeated for all five folds, and the results were averaged to deliver a comprehensive evaluation of the model's ability to generalize to unseen data. By leveraging this method, we maximize the use of the available data for both training and testing while avoiding data leakage between sets, making it particularly suited for scenarios with limited annotated datasets.

4.2 Self-training pipeline

The self-training pipeline explored in this work is a generalization of the self-training pipeline that Nunes et al. (2024a) proposed for NER in Portuguese texts of the legal domain and is shown in Figure 4.1.

Figure 4.1 – Generic self-training pipeline



Every iteration of the cross-validation follows each of the pipeline steps detailed next.

Steps 1 and 2 comprise of the training of the first classifier using the original labeled data. After that, a sampling technique is used to sample unlabeled data as shown in steps 3, 4 and 5.

Following the active sampling step, a NER classifier is applied to each sentence in the selected unlabeled data (5 to 7.b). To decide which of those machine annotated sentences should be merged to the labeled data, the average prediction confidence for the identified entities is calculated (8). If the average confidence is lower than a threshold, it is retained in the unlabeled data corpus and is not used for training; otherwise it is merged to the labeled data and removed from the unlabeled data corpus (9.a and 10). Afterwards, the pipeline restarted using the new training set (labeled data + pseudo-labeled data) to train a new model and repeat the entire pipeline.

The stop conditions (7.a) are as follows:

- Average F1-score not incremented by at least a specific threshold defined in the hyperparameters;
- No data were added to the training;
- All available unlabeled data were utilized (nothing to sample from).

4.3 Active Sampling

This is the main point of this work and is the fourth step of the self-training pipeline. Two random-based baselines were tested against three novelty BM25 based strategies. Since BM25 relies on exact matches, the stemmed version of the sentences were used as an attempt of boosting the BM25 score of sentences that did not share the exact same words but shared the same radicals.

Here's a description of each sampling strategy that was analyzed in this work, given a sampling fetch size k.

4.3.1 Baseline

The baseline of sampling is Random Sampling, in which, out of all remaining unlabeled data, choose k samples randomly.

4.3.2 Proposed

The three proposed active sampling strategies are based in BM25 and are aware of the category distribution of the labeled data and consider the stemmed version of the sentences.

The strategies do not discriminate the quantity of each named entity in the labeled sentence. This means that for a labeled sentence to be considered as a specific category, it must contain at least one named entity of that specific category.

For example, the sentence: "João da Silva e Maria da Silva foram para Porto Alegre" contains two named entities of category person (João da Silva and Maria da Silva) and one named entity of the category location (Porto Alegre). This sentence is considered for active sampling for both categories since it contains at least one named entity of that category.

The steps to populate the sample set are described in Algorithm 1:

Algorithm 1 Sample Set Population

```
1: $Final\ sample \leftarrow \{\}$
2: for $Category\ in\ Possible\ Categories$ do
3:     $Documents \leftarrow Unlabeled\ Sentence$
4:     $Query \leftarrow Labeled\ data\ that\ contains\ Category$
5:     $X \leftarrow BM25(Documents, Query)$
6:     $Unlabeled\ Dataset \leftarrow Unabeled\ Dataset\ - X$
7:     $Final\ Sample \leftarrow Final\ Sample\ + X$
8: end for
```

Line 1 simply starts the final sample set as empty. Lines 2-8 describe the sampling process for each category: the Documents variable receives either the stemmed version of the sentences in the unlabeled dataset, and the query is the subset of stemmed sentences in the labeled dataset that contain at least one entity of the matching category. After that, the BM25 algorithm is used to return the most relevant sentences from the unlabeled data given the query. Finally, the result from BM25 is deleted from the unlabeled dataset that is being sampled and is added to the final sample set.

4.3.2.1 Proportional Categories

Sample unlabeled data following the category distribution of the labeled data. The intuition behind this approach is: the labeled data follow this specific distribution, therefore the unlabeled data will also follow this specific distribution. It is important to sample unlabeled data proportionally to learn the most relevant patterns.

4.3.2.2 Disproportional Categories

The distribution of categories will be adjusted such that the most frequent category is swapped with the least frequent category, the second most frequent category is swapped with the second least frequent category and so on. Given this reversed category distribution, sample the unlabeled data accordingly. The intuition behind this approach is: the minority categories will be hard to learn since there are few examples of them, therefore, they must be prioritized.

Example: given a labeled dataset consisting of 20% named entities of category A, 30% named entities of category B and 50% named entities of category C, the resulting sample set will be made of: 20% sentences that contain at least one named entity of category C, 30% sentences that contain at least one named entity of category B and 50% sentences that contain at least one named entity of category A.

4.3.2.3 Uniform Categories

Given k categories, sample the unlabeled data so that the resulting sample set consists of $1/k$ sentences of each category. The intuition behind this approach states that every category should be equally prioritized as they are equally hard to learn.

5 EXPERIMENTS

This section outlines the experiments conducted in this study in order to fulfil the proposed methodology. We begin by providing an overview of the Ulysses-NER-Br corpus, followed by a description of the unlabeled corpus, which consists of summaries of bills from the Brazilian Chamber of Deputies (BCoD) spanning from 1991 to 2022. This unlabeled corpus played a key role in the self-training phase, helping to expand the training data. We also detail the preprocessing steps applied to the data. Next, we introduce BERTimbau Base and the reasoning to use it in the study, and explain the sampling strategies that were implemented and used in the pipeline.

5.1 Ulysses-NER-Br corpus

The original Ulysses-NER-Br (ALBUQUERQUE et al., 2022) is a corpus in Brazilian Portuguese that contains two sources of information and is divided into two corpora, one for each reference source. The first corpus contains 9,526 sentences from 150 bill drafts from the Brazilian Chamber of Deputies, while the second contains 790 sentences from work requests.

The UlyssesNER-Br corpus is divided into two types of entities: category and type. The categories include five traditional entities: "PESSOA", "DATA", "ORGANIZAÇÃO", "EVENTO", and "LOCALIZAÇÃO". Additionally, it includes "FUNDAMENTO" and "PRODUTODELEI" as references to legislative entities. The types, in turn, are specializations of the categories, such as "PRODUTOsistema", "PRODUTOprograma", and "PRODUTOoutros" as particularizations of the "PRODUTODELEI" category.

Only the categories are considered for this work since there were not significant changes in performance whether is types or categories in the original paper.

Finally, since the original version had a data leakage problem (NUNES et al., 2024b), we are using the filtered version of the dataset. The updated version of Ulysses-NER-Br, which was used in this work, can be found on Github[1] and the difference between the original and the filtered versions of the dataset is shown in Table 5.1

[1] https://github.com/ulysses-camara/ulysses-ner-br/tree/main/PL-corpus_v2

Table 5.1 – Ulysses-NER-Br filtered

	Entities			Sentences		
Class	**#Original**	**#Filtered**	**∆**	**#Original**	**#Filtered**	**∆**
DATA	603	427	-176	522	346	-176
EVENTO	23	23	0	21	21	0
FUNDAMENTO	721	716	-5	522	519	-3
LOCAL	615	607	-8	325	319	-6
ORGANIZACAO	610	598	-12	469	460	-9
PESSOA	861	847	-14	545	539	-6
PRODUTODELEI	330	319	-11	277	267	-10
Summation	3763	3537	-226	2681	2468	-213

5.2 BCoD Bills' Summaries

The unlabeled data used in this work comes from BCoD bills' summaries and was elaborated as part of the research done by Nunes (2023). It is also publicly available as HuggingFace[2] dataset.

5.3 Data preprocessing

Firstly, the bill's summaries were split into sentences using spaCy[3], an open-source library for Natural Language Processing in Python. Out of 155,710 original sentences, we end up with 172,886 after splitting them.

Finally, all duplicate sentences were excluded and the remaining ones that were already present in the UlyssesNER-Br corpus are excluded. This is important to avoid contamination and overfitting, as reported in Nunes et al. (2024a). The final set contains 164,453 of unlabeled sentences.

5.4 BERTimbau Base

The aim of this work is to evaluate different sampling strategies, not to establish a benchmark for model performance. Therefore, BERTimbau Base was chosen over the larger BERTimbau Large. For context, BERTimbau Base has 12 layers and 110 million

[2] https://huggingface.co/datasets/ronunes/LegiSubject-Br-Summaries

[3] https://spacy.io/

parameters, compared to BERTimbau Large's 24 layers and 335 million parameters.

So, even though it is possible that larger models, such as BERTimbau Large, could provide better results, the difference between training times did not justify using a larger model for this work.

It is also important to note that even though BERTimbau Base is not as large as other models, its performance for NER has been proved to be competitive when compared to other language-specific models as reported by Nunes et al. (2024a).

5.5 Setup

The setup consisted of a computer with an Intel i7-14700 CPU, 16.0 GB of memory and with an Nvidia GeForce RTX 4060 GPU, which was used for training and evaluating the models.

Python 3.11 was selected for its compatibility with the libraries used in this project. The HuggingFace Trainer API[4] was utilized to train the bert-base-portuguese-cased BERT model (BERTimbau Base) for the NER task. Unlabeled data from the BCoD bill summaries was split into sentences using spaCy[5], while NLTK[6] was used to stem sentences in both the labeled and unlabeled datasets. Finally, seqeval[7] library was employed to evaluate the NER model's performance, offering specialized metrics such as precision, recall, and F1 score tailored for sequence labeling tasks. The graphs displaying the results were generated using Matplotlib[8].

5.5.1 Model Hyperparameters

Following the work proposed in the original self-training pipeline (NUNES et al., 2024a), the following hyperparameters to build the model were used: learning_rate = 2e-05, weight_decay = 0.01, epsilon = 1e-08 and num_epochs = 10. The remaining non-specified parameters follow the model's default parameters. Truncation and padding were applied to ensure sentences matched the maximum length of 512 tokens.

[4] https://github.com/ulysses-camara/ulysses-ner-br/tree/main/PL-corpus_v2

[5] https://spacy.io

[6] https://www.nltk.org

[7] https://github.com/chakki-works/seqeval

[8] https://matplotlib.org

5.5.2 Self-training Hyperparameters

In total, 4 active sampling strategies were tested:

- Random
- BM25 using Proportional Categories
- BM25 using Disproportional Categories
- BM25 using Uniform Categories

With a combination of different hyperparameters: sampling size, F1 score patience, minimum F1 increase and average prediction confidence.

The different sampling fetch sizes used in this study were: 500, 2500 and 5000.

F1 score patience, the parameter responsible for setting the maximum amount of times the iteration can be retried after not increasing the overall F1, is equal to 4 and a minimum increase of 0.005 in overall F1 must be achieved in the iteration.

The average prediction confidence threshold we used was 0.99, which was the one that showed the best results in the original self-training pipeline (NUNES et al., 2024a).

Random-based sampling might not contain suitable examples for training, resulting in no additions to the training set. To address this, a waiting criterion that allows for a maximum of W new samplings (using different random seeds) before terminating the self-training process. This work used this sample patience parameter as 5.

5.5.3 Results and discussion

Table 5.2 shows the mean and standard deviation of the F1-scores using 5-fold cross-validation from different combinations of fetch sizes and sampling strategies. The results indicate that the proposed sampling strategies deliver comparable overall performance to the baseline when using the same fetch size.

However, it is necessary to take into account not only the sampling strategy and sampling fetch sizes, but also, how many of those pre-selected samples passed the minimum confidence threshold. Figure 5.1 shows the average size of the training data to highlight that despite having the same fetch sizes, the Random Sampling strategy managed to select more samples that were effectively added to the training data throughout the iterations across different folds.

This discrepancy can be attributed to BM25's process of selecting the most rele-

Table 5.2 – Cross-validation results to each sampling strategy with different sampling fetch sizes

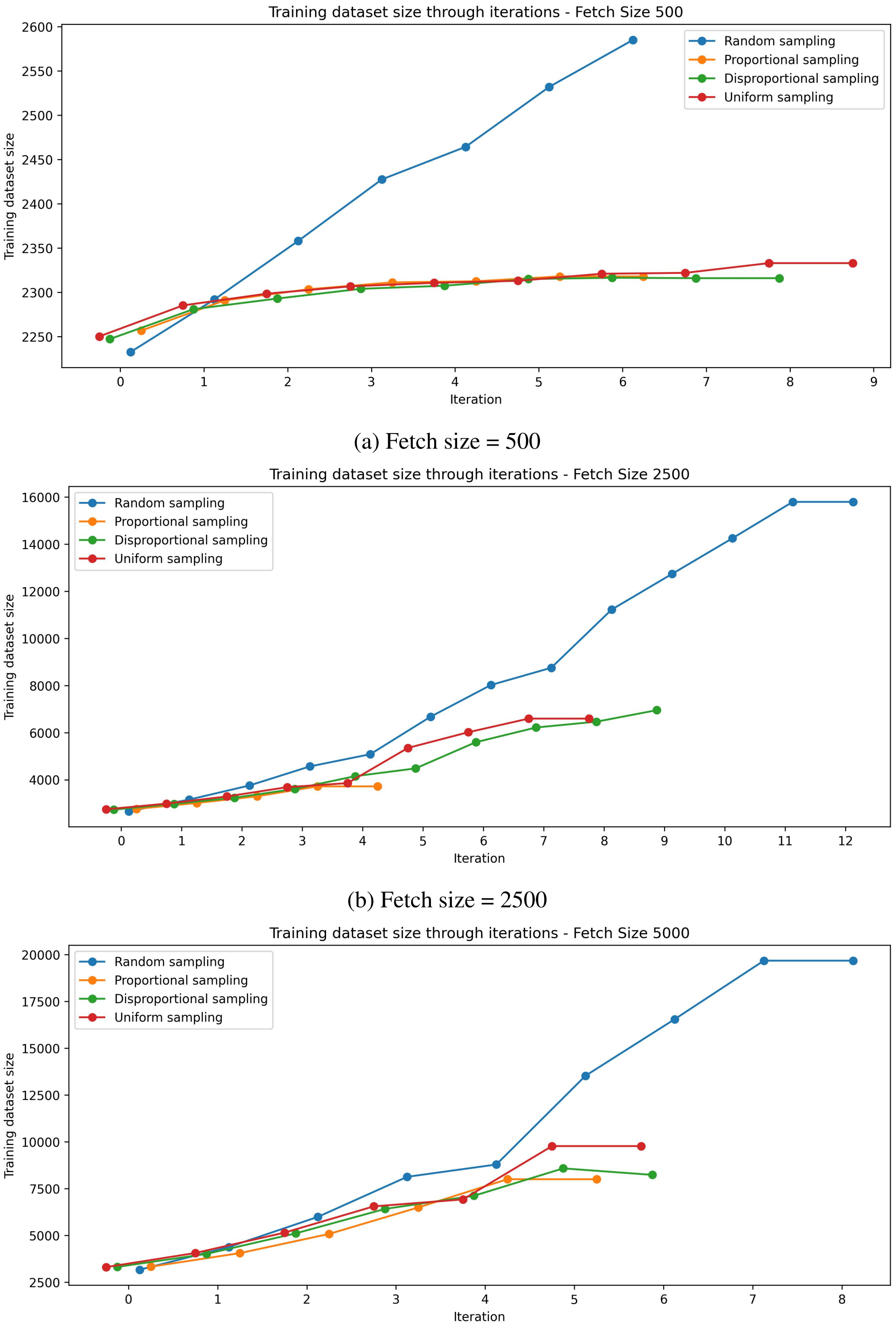
Fetch Size	Overall	Data	Evento	Fundamento	Local	Organizacao	Pessoa	Produtodelei
Random								
500	**82.10 ± 0.73**	95.35 ± 2.60	69.76 ± 7.92	86.34 ± 1.58	84.07 ± 1.26	80.53 ± 3.73	88.54 ± 1.24	70.14 ± 6.58
2500	82.03 ± 1.20	95.64 ± 2.50	69.10 ± 8.78	**86.68 ± 1.58**	82.75 ± 2.78	79.43 ± 1.66	88.95 ± 1.75	71.65 ± 7.25
5000	81.93 ± 0.72	95.29 ± 2.80	**74.76 ± 15.91**	84.00 ± 5.09	84.83 ± 1.82	78.81 ± 2.98	88.12 ± 2.71	67.71 ± 7.37
Proportional								
500	80.31 ± 0.81	95.33 ± 2.69	61.29 ± 7.77	84.13 ± 3.09	83.30 ± 2.91	**80.30 ± 2.84**	87.18 ± 1.98	70.68 ± 3.96
2500	81.18 ± 0.63	94.87 ± 2.91	66.43 ± 6.32	85.42 ± 3.71	**84.91 ± 1.64**	78.03 ± 3.04	88.76 ± 2.04	69.88 ± 5.48
5000	81.23 ± 0.90	**95.80 ± 2.06**	68.86 ± 10.89	82.01 ± 2.86	84.48 ± 1.96	79.86 ± 2.04	88.23 ± 1.38	69.35 ± 4.81
Disproportional								
500	81.42 ± 0.99	95.20 ± 2.45	67.86 ± 9.78	84.80 ± 2.80	83.81 ± 1.90	78.78 ± 2.59	88.01 ± 2.57	**71.49 ± 5.64**
2500	80.61 ± 1.64	94.74 ± 3.38	65.87 ± 14.66	82.96 ± 3.84	83.73 ± 2.82	79.01 ± 3.15	88.13 ± 1.89	69.85 ± 2.69
5000	81.52 ± 1.06	94.99 ± 2.29	68.10 ± 7.41	84.87 ± 3.07	84.51 ± 2.06	78.49 ± 2.73	**89.16 ± 0.88**	70.52 ± 5.55
Uniform								
500	81.16 ± 1.41	95.02 ± 3.17	67.67 ± 11.4	85.41 ± 3.15	83.27 ± 3.05	79.61 ± 3.03	87.18 ± 1.73	70.00 ± 3.59
2500	80.92 ± 2.68	95.54 ± 2.69	66.83 ± 22.82	84.79 ± 3.87	84.34 ± 1.86	78.36 ± 5.03	87.67 ± 2.59	68.90 ± 3.99
5000	80.02 ± 2.45	94.67 ± 3.69	62.54 ± 24.00	82.81 ± 3.92	84.41 ± 2.20	78.86 ± 1.64	88.30 ± 2.43	68.59 ± 4.83

vant sentences to labeled dataset from the unlabeled dataset. These pre-selected sentences are often similar to each other, which can result in their confidence scores for identified named entities being below the minimum threshold for similar reasons. As a result, these sentences are not added to the training data and are likely to be sampled again in subsequent rounds. However, there is no guarantee they will meet the confidence threshold on subsequent attempts either. This cycle effectively reduces the sampling size for this strategy and limits the addition of new, diverse unlabeled data to the training set.

The limited amount of new data being added may contribute to the poor performance of the BM25-based strategies. Additionally, even if all the samples selected by BM25 were added to the training data in a single iteration, these samples might be so similar to the existing labeled data that they could lead to overfitting, further hindering the model's performance.

Figure 5.2 show the mean overall F1-score across the 5 folds to highlight the difference between sampling strategies. This is relevant to show that even though a sample passed the minimum confidence threshold and was added to the training data it doesn't mean that it improves the model's overall performance for the subsequent iterations.

Table 5.2 also shows that although the Disproportional Sampling strategy was intended to improve the performance of underrepresented named entity categories, such as EVENTO (which makes up around 0.65% of the named entities found in filtered Ulysses-NER-Br dataset), its F1-score for this category is similar to those achieved by other strategies. On the other hand, PESSOA, the most well-represented named entity category (representing about 24% of the named entities present in the filtered Ulysses-NER-Br dataset), exhibits minimal variation in F1-scores across different sampling strategies. These results may be influenced by the small amount of machine-annotated data added to the training



(a) Fetch size = 500

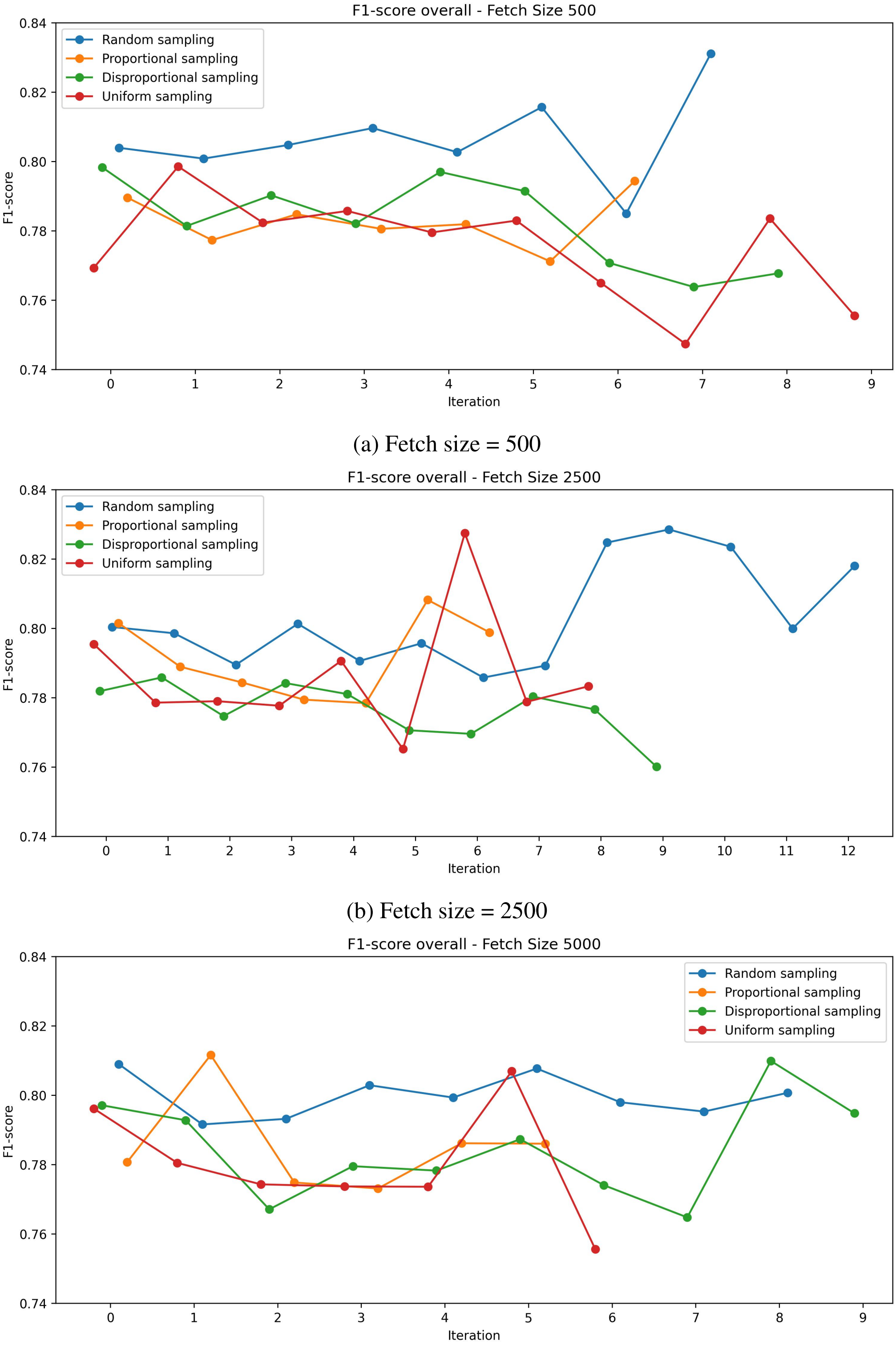
(b) Fetch size = 2500

(c) Fetch size = 5000

Figure 5.1 – F1-scores of different sampling strategies through the iterations, grouped by sampling fetch size

set during the iterations using BM25-based sampling strategies.

An interesting observation from Table 5.2 is that PRODUTODELEI which is the second most underrepresented category in the Ulysses-NER-Br dataset (accounting for



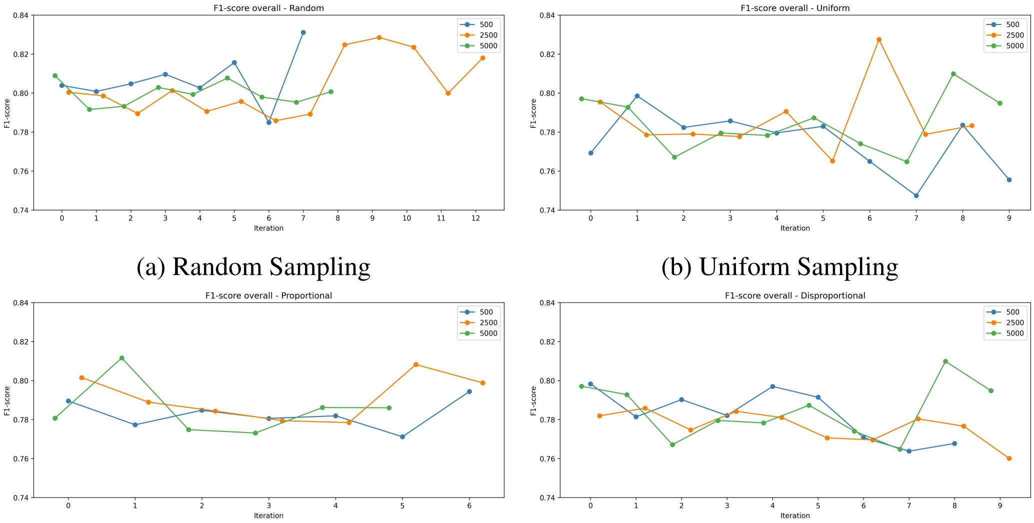
(a) Fetch size = 500

(b) Fetch size = 2500

(c) Fetch size = 5000

Figure 5.2 – F1-scores of different sampling strategies through the iterations, grouped by sampling fetch size

about 9% of the named entities in the filtered dataset), shows mean F1-scores similar to those of the EVENTO category, which is around 13 times more underrepresented. This suggests that both categories may be affected by underfitting, despite one having significantly more labeled examples than the other. The F1-score mean and standard deviation of each named entity category can be found in Appendix A.



(a) Random Sampling

(b) Uniform Sampling

(c) Proportional Sampling

(d) Disproportional Sampling

Figure 5.3 – F1-scores of different sampling fetch sizes through the iterations, grouped by sampling strategy

Figure 5.3 presents the mean overall F1-score across the five folds, highlighting the impact of different fetch sizes. The purpose of these graphs is to demonstrate that a larger fetch size does not necessarily lead to better performance. For example, Random Sampling with a fetch size of 500 achieves a higher overall F1-score compared to fetch sizes of 2500 or 5000. Conversely, smaller fetch sizes do not always result in better model performance, as illustrated by Figure 5.3d.

6 CONCLUSION

In summary, the results obtained with the novelty sampling strategies, with the hyperparameters that were chosen, achieved performance very close to random sampling. The analysis of Figures 5.1, 5.2 and 5.3 emphasize the importance of analyzing several factors other than simply the sampling strategy and the sampling fetch size, such as: how many of the pre-selected samples were actually added to the training dataset, the diversity of these samples in relation to each other and the current training data, and how representative the selected samples are of the overall unlabeled dataset.

Using BM25 to sample from the unlabeled data and applying a confidence threshold may have limited the performance of the proposed strategies. High-confidence samples could be too similar to both the existing training set and to each other, offering little diversity. This lack of variety could ultimately hinder the model's ability to generalize effectively. To remedy this situation, a straightforward solution could be setting the minimum confidence level high enough to avoid adding noisy data, yet low enough to ensure a significant amount of machine-annotated data is incorporated into the training set during each iteration. To do so would require the fine-tuning of the threshold hyperparameter, though.

Settles (2012) emphasizes that the diversity of samples is crucial to avoid redundancy and to ensure that the model learns to generalize across various, potentially more complex, scenarios, rather than merely focusing on a limited aspect of the data. Furthermore, it is noted that the diversity of the samples in relation to the training data is important for preventing overfitting to the labeled data. By doing so, it encourages the model to explore underrepresented areas of the feature space, rather than reinforcing the categories or examples that are already well-covered. Therefore, both diversities play a role in active sampling.

An idea would be to use IR to sample the data, with one of the proposed strategies, for example. With this pre-selection in hand, then use diversity-based sampling such as in Tran et al. (2017), Chen et al. (2015) to select the most dissimilar samples among themselves.

In essence, future work should take in consideration both types do diversity while also making sure that enough non-redundant high-confidence machine-annotated data is being added to the training dataset throughout the iterations.

REFERENCES

ALBUQUERQUE, H. O. et al. Ulyssesner-br: A corpus of brazilian legislative documents for named entity recognition. In: PINHEIRO, V. et al. (Ed.). **Computational Processing of the Portuguese Language**. Cham: Springer International Publishing, 2022. p. 3–14. ISBN 978-3-030-98305-5.

ALBUQUERQUE, H. O. et al. Named entity recognition: a survey for the portuguese language. **Procesamiento del Lenguaje Natural**, 2023.

ANGELIDIS, I.; CHALKIDIS, I.; KOUBARAKIS, M. Named entity recognition, linking and generation for greek legislation. In: **International Conference on Legal Knowledge and Information Systems**. [s.n.], 2018. Available from Internet: <https://api.semanticscholar.org/CorpusID:55699546>.

ANTOUN, W.; BALY, F.; HAJJ, H. AraBERT: Transformer-based model for Arabic language understanding. In: AL-KHALIFA, H. et al. (Ed.). **Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection**. Marseille, France: European Language Resource Association, 2020. p. 9–15. ISBN 979-10-95546-51-1. Available from Internet: <https://aclanthology.org/2020.osact-1.2/>.

CAÑETE, J. et al. **Spanish Pre-trained BERT Model and Evaluation Data**. 2023. Available from Internet: <https://arxiv.org/abs/2308.02976>.

CHALKIDIS, I. et al. **LEGAL-BERT: The Muppets straight out of Law School**. 2020. Available from Internet: <https://arxiv.org/abs/2010.02559>.

CHEN, Y. et al. A study of active learning methods for named entity recognition in clinical text. **Journal of Biomedical Informatics**, v. 58, p. 11–18, 2015. ISSN 1532-0464. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1532046415002038>.

DARJI, H.; MITROVIĆ, J.; GRANITZER, M. German bert model for legal named entity recognition. In: **Proceedings of the 15th International Conference on Agents and Artificial Intelligence**. SCITEPRESS - Science and Technology Publications, 2023. p. 723–728. Available from Internet: <http://dx.doi.org/10.5220/0011749400003393>.

DEVLIN, J. et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. 2019. Available from Internet: <https://arxiv.org/abs/1810.04805>.

DOZIER, C. et al. Named entity recognition and resolution in legal text. In: ____. **Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 27–43. ISBN 978-3-642-12837-0. Available from Internet: <https://doi.org/10.1007/978-3-642-12837-0_2>.

DUPRE, R. et al. Improving dataset volumes and model accuracy with semi-supervised iterative self-learning. **IEEE Transactions on Image Processing**, v. 29, p. 4337–4348, 2020.

Fraile Navarro, D. et al. Clinical named entity recognition and relation extraction using natural language processing of medical free text: A systematic review. **International Journal of Medical Informatics**, v. 177, p. 105122, 2023. ISSN 1386-5056. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1386505623001405>.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 2nd. ed. Sebastopol, USA: O'Reilly Media, 2019.

HIRSCHBERG, J.; MANNING, C. D. Advances in natural language processing. **Science**, v. 349, n. 6245, p. 261–266, 2015. Available from Internet: <https://www.science.org/doi/abs/10.1126/science.aaa8685>.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing**. 2nd. ed. [S. l]: Prentice Hall, 2008.

LEE, D.-H. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. **ICML 2013 Workshop : Challenges in Representation Learning (WREPL)**, July 2013.

LI, B.; HOU, Y.; CHE, W. Data augmentation approaches in natural language processing: A survey. **AI Open**, Elsevier BV, v. 3, p. 71–90, 2022. ISSN 2666-6510. Available from Internet: <http://dx.doi.org/10.1016/j.aiopen.2022.03.001>.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to Information Retrieval**. Cambridge, UK: Cambridge University Press, 2008.

MOLLÁ, D.; ZAANEN, M. van; SMITH, D. Named entity recognition for question answering. In: CAVEDON, L.; ZUKERMAN, I. (Ed.). **Proceedings of the Australasian Language Technology Workshop 2006**. Sydney, Australia: [s.n.], 2006. p. 51–58. Available from Internet: <https://aclanthology.org/U06-1009>.

NATH, N.; LEE, S.-H.; LEE, I. Near: Named entity and attribute recognition of clinical concepts. **Journal of Biomedical Informatics**, Elsevier BV, v. 130, p. 104092, jun. 2022. ISSN 1532-0464. Available from Internet: <http://dx.doi.org/10.1016/j.jbi.2022.104092>.

NUNES, R. O. **A Classification Approach for Estimating Subjects of Bills in the Brazilian Chamber of Deputies**. Bachelor's thesis — Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil, 2023. Accessed: 2025-01-03. Available from Internet: <https://lume.ufrgs.br/bitstream/handle/10183/267612/001188065.pdf?sequence=1>.

NUNES, R. O. et al. A named entity recognition approach for Portuguese legislative texts using self-learning. In: GAMALLO, P. et al. (Ed.). **Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1**. Santiago de Compostela, Galicia/Spain: Association for Computational Lingustics, 2024. p. 290–300. Available from Internet: <https://aclanthology.org/2024.propor-1.30>.

NUNES, R. O. et al. Reconhecimento de entidades nomeadas e vazamento de dados em textos legislativos: Uma reavaliação da literatura. In: . [s.n.], 2024. Available from Internet: <http://dx.doi.org/10.13140/RG.2.2.25781.69602>.

NUNES, R. O. et al. An evaluation of large language models for geological named entity recognition. **ResearchGate**, 2025. Available at: <https://www.researchgate.net/publication/383822506_An_Evaluation_of_Large_Language_Models_for_Geological_Named_Entity_Recognition>.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.

SETTLES, B. Active learning. In: **Synthesis Lectures on Artificial Intelligence and Machine Learning**. [S. l]: Springer Cham, 2012. v. 4.

SHAH, A. et al. **FiNER-ORD: Financial Named Entity Recognition Open Research Dataset**. 2024. Available from Internet: <https://arxiv.org/abs/2302.11157>.

SHEN, Y. et al. **Deep Active Learning for Named Entity Recognition**. 2018. Available from Internet: <https://arxiv.org/abs/1707.05928>.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: Pretrained bert models for brazilian portuguese. In: CERRI, R.; PRATI, R. C. (Ed.). **Intelligent Systems**. Cham: Springer International Publishing, 2020. p. 403–417. ISBN 978-3-030-61377-8.

TRAN, V. C. et al. A combination of active learning and self-learning for named entity recognition on twitter using conditional random fields. **Knowledge-Based Systems**, v. 132, p. 179–187, 2017. ISSN 0950-7051. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0950705117303040>.

VASWANI, A. et al. Attention is all you need. **CoRR**, abs/1706.03762, 2017. Available from Internet: <http://arxiv.org/abs/1706.03762>.

YUAN, Z.; ZHANG, H. Improving named entity recognition of chinese legal documents by lexical enhancement. In: **2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)**. [S.l.: s.n.], 2021. p. 999–1004.

ZHU, X.; GOLDBERG, A. Introduction to semi-supervised learning. In: **Synthesis Lectures on Artificial Intelligence and Machine Learning**. [S. l]: Springer Cham, 2009. v. 2.

APPENDIX A — EXTENDED RESULTS

In this appendix, the mean F1-score and its standard deviation obtained for the named entities are reported through bar charts. The goal of these graphic representations is to offer a more comprehensive overview of the results.

Figure A.1 – F1-score of the DATA category

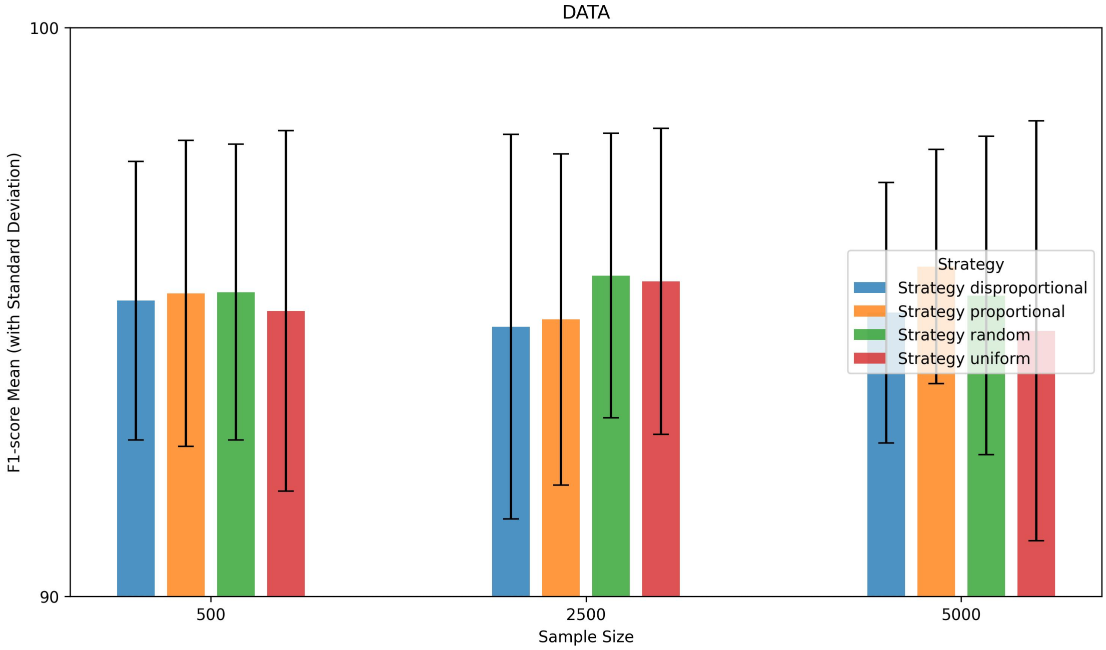


Figure A.2 – F1-score of the EVENTO category

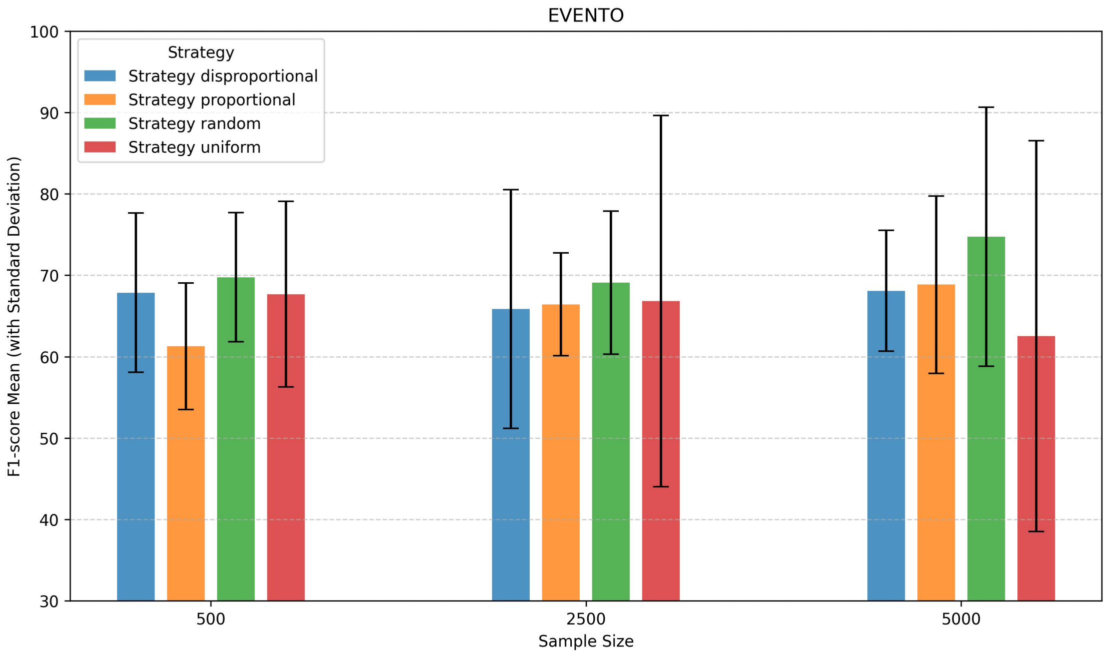


Figure A.3 – F1-score of the FUNDAMENTO category

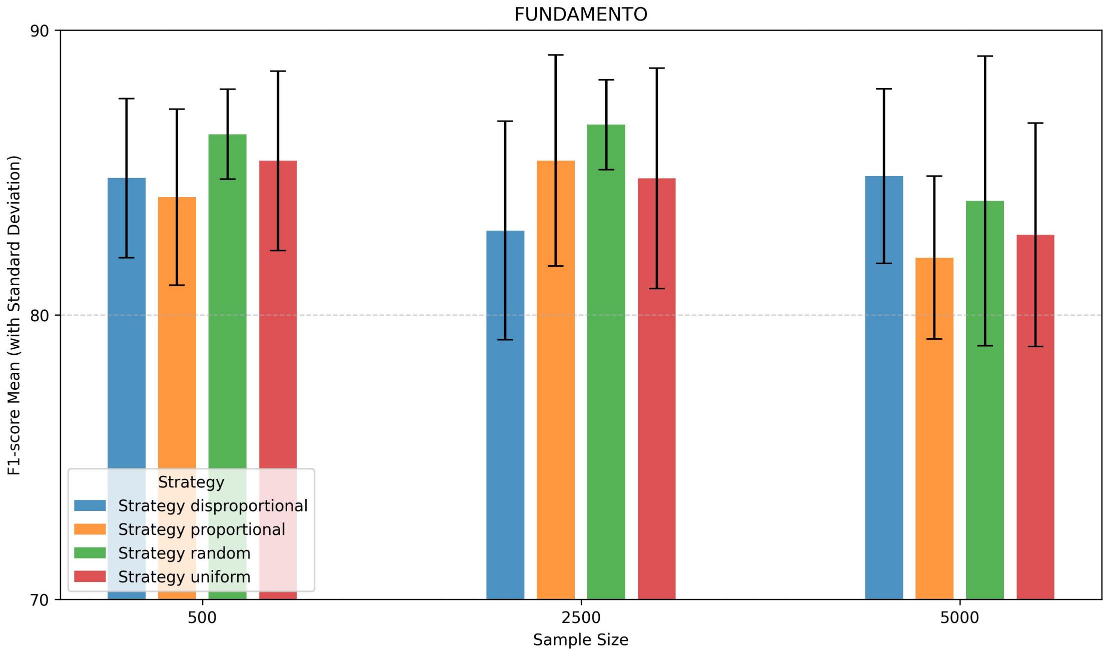


Figure A.4 – F1-score of the LOCAL category

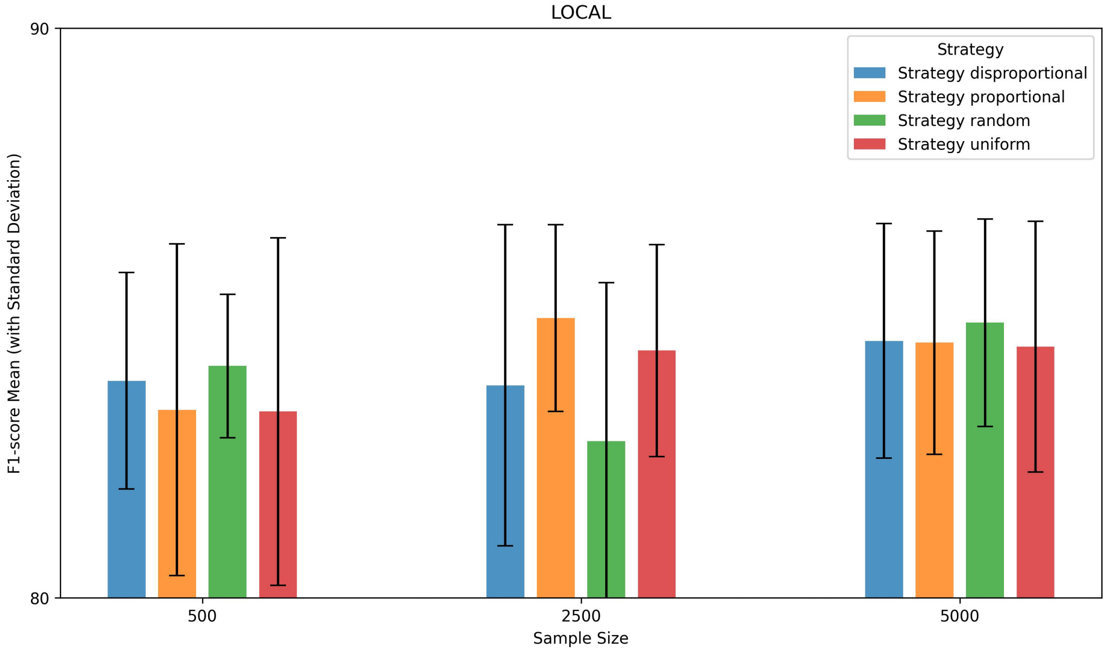


Figure A.5 – F1-score of the ORGANIZACAO category

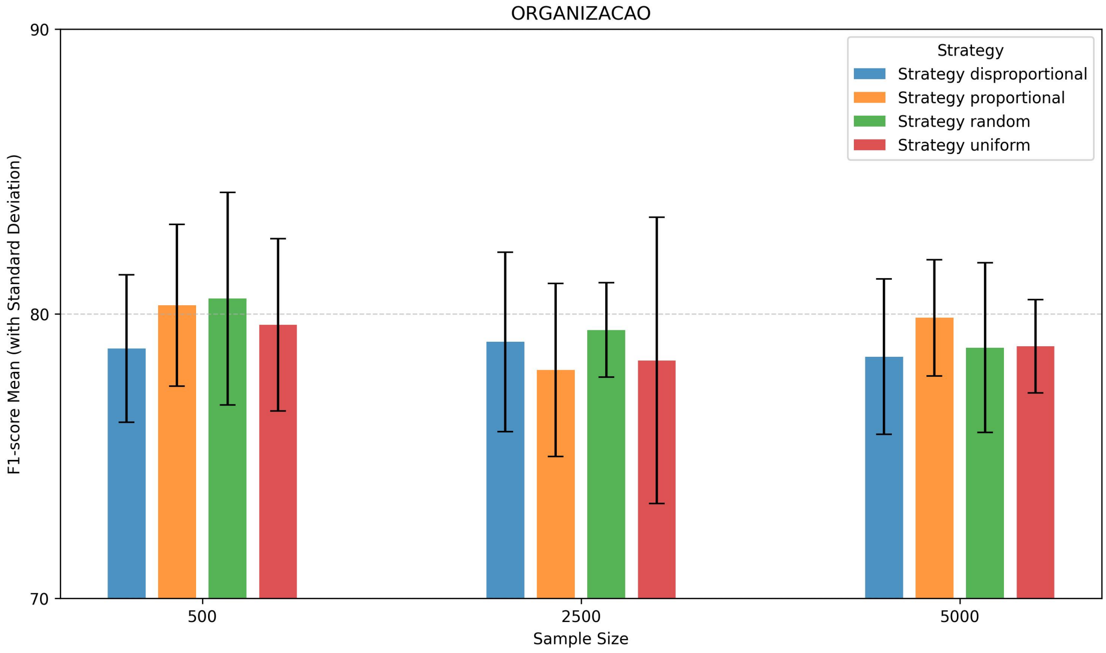


Figure A.6 – F1-score of the PESSOA category

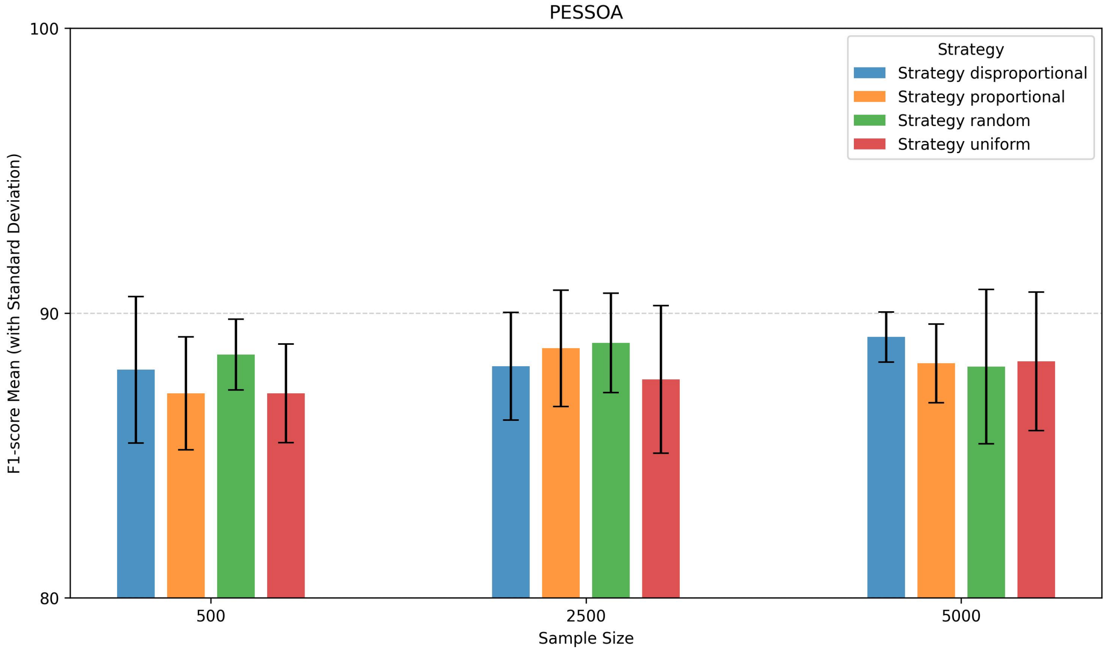


Figure A.7 – F1-score of the PRODUTODELEI category

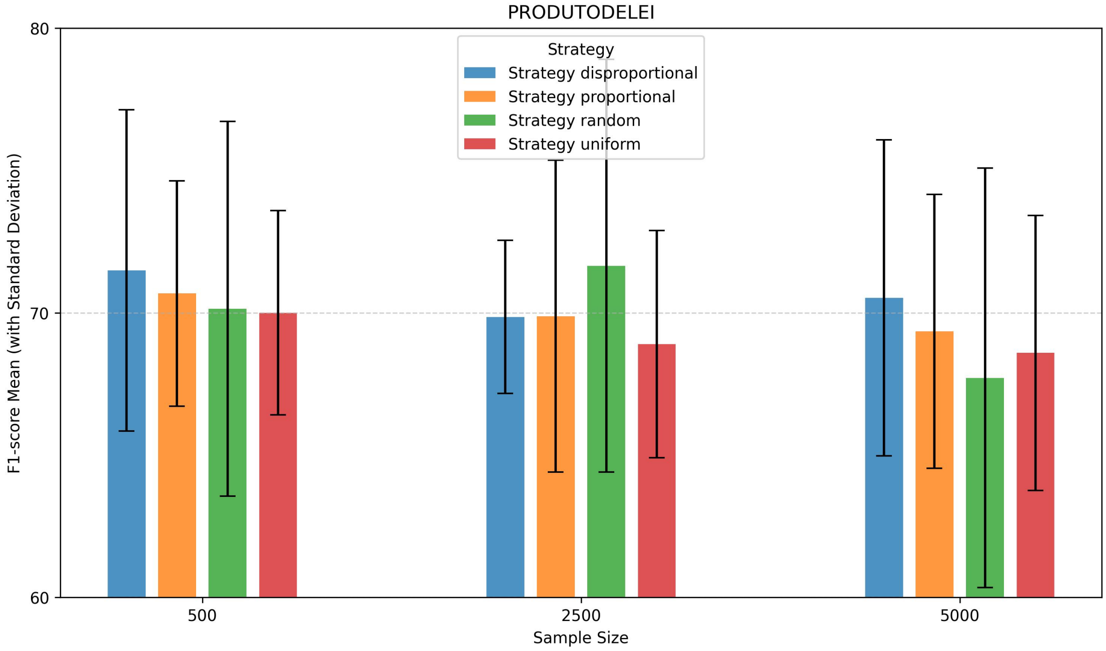
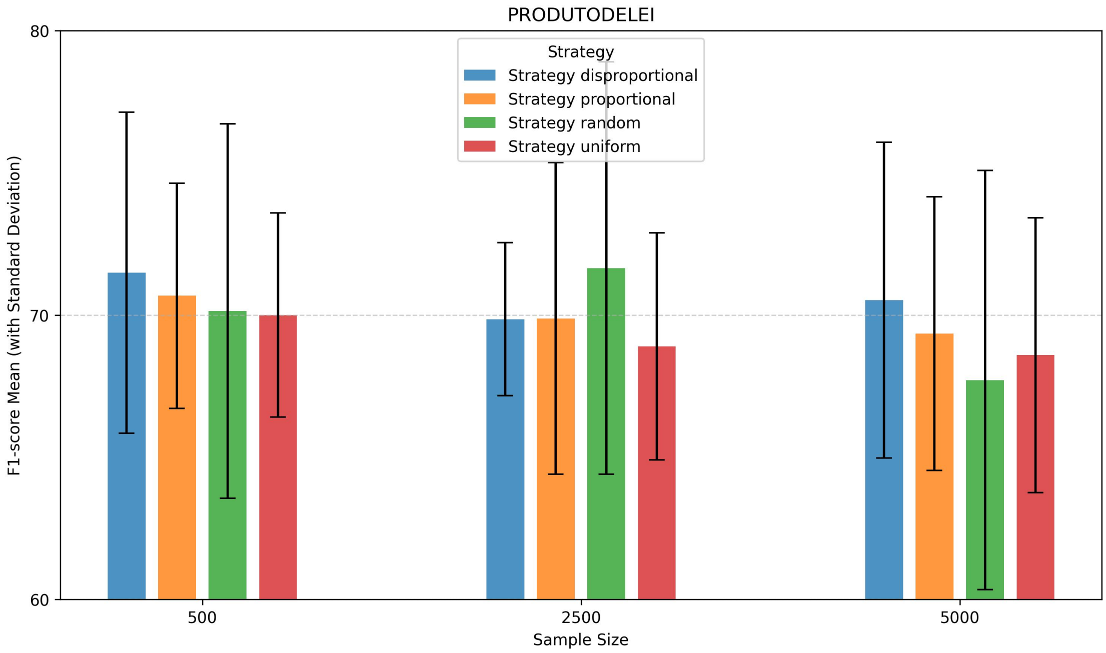


Figure A.8 – F1-score of the PRODUTODELEI category


PRODUTODELEI
Strategy
Strategy disproportional
Strategy proportional
Strategy random
Strategy uniform
F1-score Mean (with Standard Deviation)
Sample Size