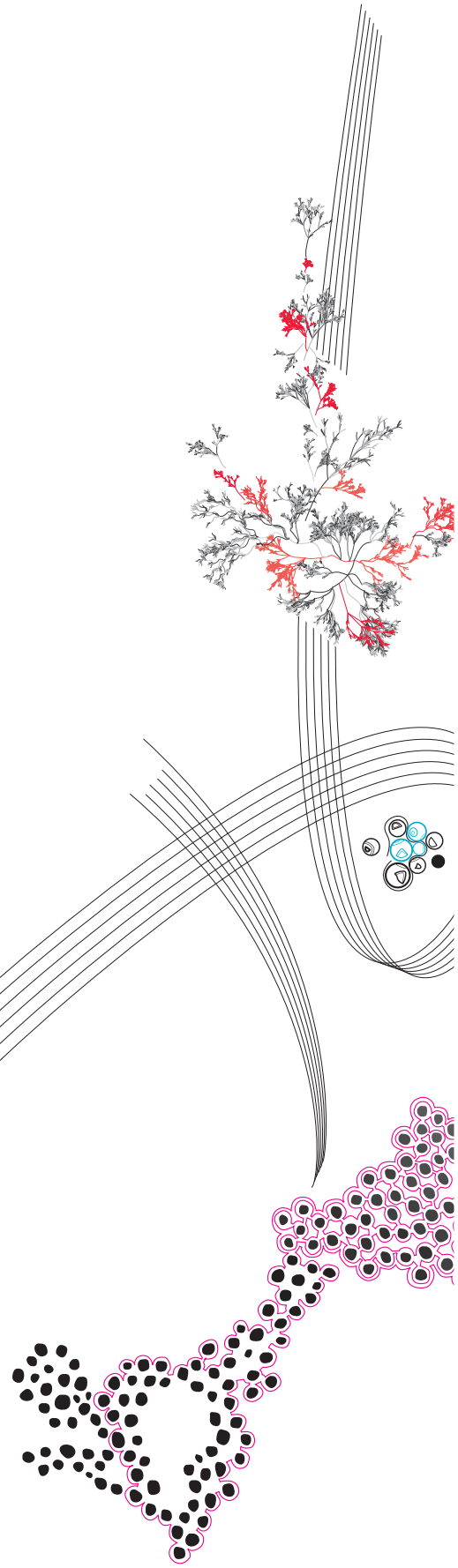


MSc Interaction Technology
Final Project

Cold-start Active Learning for Text Classification of Business Documents

Bachir Kaddis Beshay Amir

Supervisor: Dr. Ing. G. Englebienne
Supervisor: Dr. E. Mocanu
Business Supervisor: F. Visalli
Business Collaborator: A. Lanza
Business Collaborator: P. Papaleo

September 27, 2023

MSc Interaction Technology
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Abstract

Recent years have witnessed a remarkable surge in Artificial Intelligence (AI), permeating diverse domains and streamlining tasks for enhanced efficiency. Within this landscape, the realm of Natural Language Processing (NLP) has garnered significant attention, offering the promise of automating tasks involving human language. This thesis is propelled by a profound interest in the practical applications of AI and NLP, particularly within industrial contexts where unlabeled data is abundant yet laborious to annotate. The study focuses on Active Learning, with a distinctive emphasis on its 'cold start' phase, a scenario common in real-world applications where limited or no labels are available. Active Learning, a specialized branch of machine learning, goes beyond conventional training by selecting the most informative data points for labeling, operating under the premise that strategic data selection can lead to superior performance with fewer training instances. This is particularly advantageous when abundant unlabeled data exists, but labeling is a resource-intensive endeavor. By intelligently interacting with a human expert, referred to as an oracle, an active learner acquires true labels for select data points, with the goal of minimizing labeling efforts without compromising learning efficacy.

The thesis centers on two pivotal phases of active learning: the 'cold start' and the subsequent 'warm start'. The quality of the initial pool of labeled data, often referred to as the 'cold start' phase, significantly influences the efficiency and accuracy of ensuing learning iterations. However, this critical phase remains underexplored, particularly in the context of text classification. The study aims to bridge this knowledge gap, focusing on techniques that can judiciously construct an initial labeled pool to enable more effective sampling decisions in later iterations, ultimately optimizing the active learning process.

Moreover, this research holds paramount relevance in the contemporary technological landscape. For instance, in the case of Altilia, a company specializing in AI-driven intelligent document processing, the intelligent selection of instances for labeling during the early stages of active learning is of paramount importance given the cost associated with labeling documents.

The study seeks to answer two fundamental research questions: Can cold start techniques enhance subsequent active learning iterations? When do warm start techniques outperform their cold start counterparts? The investigation is driven by the hypothesis that while cold start techniques excel in the early stages, warm start techniques, leveraging uncertainty measures, eventually supersede them. Nevertheless, optimizing the initial sample selection holds potential for significant process enhancements.

The thesis significantly advances the domain of active learning, with a specific focus on the initial 'cold start' phase in text classification. It introduces a pioneering methodology for

gauging the efficacy of cold start techniques and establishes an experimental framework for rigorous comparative analysis. Moreover, the thesis introduces three innovative cold start techniques, broadening the spectrum of available methodologies in active learning. These contributions collectively underscore the notion that substantial progress is often the result of incremental advancements.

Contents

Chapter 1

Introduction

1.1 Motivation

Artificial Intelligence (AI) has grown remarkably in recent years, finding applications in various domains that aim to simplify tasks and improve efficiency. Among these applications, Natural Language Processing (NLP) has garnered significant interest, particularly due to its potential to automate repetitive tasks involving human language. This thesis is inspired by a keen interest in the practical applications of AI and NLP, particularly within an industrial setting where unlabeled data is abundant but costly to label. We focus on the concept of Active Learning, with a special emphasis on its 'cold start' phase, where little to no labels are available, a common and challenging scenario in many real-world applications.

Machine learning (ML), a category of algorithms within AI, fundamentally involves training a model using a dataset so that the model can make accurate predictions or decisions without being explicitly programmed to perform the task.

Active learning, a special case of machine learning, takes this concept a step further by selecting the most informative data points for labeling. Unlike traditional machine learning, where all instances in the training set are equally considered, active learning operates under the assumption that if a learning algorithm can choose the data it wants to learn from, it can perform better with less training.

This technique is particularly beneficial when unlabeled data is abundant but labeling it is costly or time-consuming. An active learner has a certain level of interaction with a human expert, referred to as an oracle, for obtaining the true label for selective data points. The aim of active learning is to minimize the labeling effort without compromising the learning performance. Thus, active learning can significantly enhance the efficiency of machine learning processes in various applications, particularly in scenarios where data labeling is an expensive or constrained resource.

Active learning consists of two primary phases: the cold start and the warm start. In this iterative process, the quality of the initial pool of labeled data—commonly referred to as the "cold start" stage—can significantly influence the efficiency and accuracy of subsequent learning iterations. However, the existing body of research has not yet fully explored this critical phase, especially within the context of text classification. This leaves a knowledge gap, which becomes particularly evident when addressing the limitations of uncertainty-based methods during the early stages of active learning.

The cold start problem involves the effective selection of initial data for labeling, an area that is under-researched but carries substantial implications for the overall success of active learning. Most active learning methodologies necessitate this initial pool of labeled data to jump-start the learning process, underscoring the importance of thoroughly investigating cold start strategies. Consequently, the study aims to address this problem and focus on the development of techniques that can intelligently construct an initial labeled pool. This enables more effective sampling decisions in later iterations, optimizing the active learning process and improving its performance during the warm start phase. The ultimate goal is to enhance the robustness of active learning methodologies, thereby contributing to the progression of text classification practices.

Moreover, in the contemporary technological landscape, this research carries paramount importance. For instance, Altilia, a company operating in the field of artificial intelligence applied to intelligent document processing, necessitates labeled data to train its information extraction models. Given the cost of labeling documents, intelligent selection of instances for labeling during the early stages of active learning becomes crucial.

1.2 Research Questions and Contributions

This research aims to address the problem of improving the active learning process by investigating the cold start phase, comparing different cold start techniques, and determining the threshold where warm start techniques outperform the cold start ones. The research questions guiding this study are twofold:

1) Can cold start techniques provide an initial labeled pool that enhances the subsequent active learning iterations? In essence, an initial labeled pool with specific characteristics may facilitate the warm start techniques in selecting more useful examples and help mitigate the sampling bias.

2) When do warm start techniques become more effective than the cold start ones? As the size of labeled samples increases, the reliability of uncertainty measures used by warm start techniques improves. At a certain point, it is expected that warm start techniques will surpass the cold start ones, but the question remains - when exactly?

The hypothesis driving this investigation is that while cold start techniques will be more effective at the beginning of the active learning process, they will be superseded by warm start ones later on, given their exploitation of uncertainty measures. Still, an optimized selection of the initial samples may bring significant improvements to the overall process.

This thesis presents significant contributions to the field of active learning, particularly focusing on the cold start phase in text classification. Firstly, it provides an innovative approach to assess the performance of cold start active learning techniques. This unique perspective offers a fresh lens to appraise their effectiveness, paving the way for a better understanding and further enhancement of these methods.
Secondly, the study devises an experimental framework that allows for a comparative analysis of cold and warm start techniques. By studying the performance crossover point between these techniques, the proposed experimental setup serves as a critical tool in determining the optimal application and transition points for each method.

Finally, the thesis introduces the development of three novel cold start techniques for text classification. These methods not only expand the repertoire of active learning techniques but also hold promise for improved performance in the early stages of active learning, particularly in real-world, industrial settings.

1.3 Approach

To answer the research questions, two experiments were conducted. In the first one, different cold start techniques were compared based on the efficacy of the constructed initial labeled dataset. The experiment is designed in such a way that the only variable between the different active learning processes is the initial batch from which it starts from. Each of the proposed methods is optimized to select a sample of examples with certain characteristics: label diversity and typical data selection. In the second experiment, these techniques were compared amongst each other and with a warm start technique, BADGE, in active learning iterations. The setting of the second experiment is standard AL cycles but at each iteration two labeled sets are created. The labeled sets differ only for the last added batch: in one the batch is selected by the cold start technique in the other by the warm start one. In such way it is verified when the warm start method is able to select more informative sample batches.

Two datasets were used in the experiments - Altilia's dataset and the AGNews dataset. The choice was driven by the need for diverse domains and comparability with academic datasets. Both datasets were manipulated for to have better comparability: their sizes were reduced and only the four major classes were kept.

Several cold start techniques were tested - T-CALR (Textual Cold start Active Learning through Representative sampling), Iterative T-CALR, BCS (Balanced Cold Start), alongside baselines like random sampling and simulated balanced sampling. These techniques work with no or few labeled samples. T-CALR is based on the sentence embeddings using SBERT models, clustering the space and representative data selection. It makes no use of labeled data and allows for one shot sampling. Iterative T-CALR and BCS, on the other hand, make use of labels. In the first case by fine-tuning the SBERT model with the SetFit contrastive fine-tuning paradigm and in the second by accounting for class balance given the class distribution in the current labeled set. Each technique was chosen with specific considerations, ranging from being inspired by existing literature, exploiting label information, to understanding how class balance affects performance.

BADGE, a warm start technique, was selected for comparison due to its prominence in the literature.

1.4 Results

The results of the first experiment, performed on Altilia's data, indicate that the presence of typical data but most importantly class balance in the initial set can significantly influence the effectiveness of active learning in its early stages. Their impact decreases as active learning progresses. However, it is found that simulating complete balance sampling achieves the best results through all active learning iterations, although none of the tested techniques could replicate these results entirely. The second experiment is conducted on 2 datasets from different domains but similar class distributions. The performance of the

tested methods and baselines are not in complete accordance in the two cases. However, the result curves indicate how Iterative T-CALR significantly outperforms T-CALR throughout the active learning cycles. This demonstrates how useful is to exploit the, even little, amount of labeled examples available.

1.5 Thesis outline

This thesis is structured as follows: The Background Theory chapter presents natural language processing foundation, state of the art methods in active learning, with a focus on the cold start literature and the research gaps. The Methodology chapter describes the data used, the preliminary work carried out, and the proposed methods. The Experiment chapter provides details about the experiment implementation and discusses the results. Finally, the Conclusion chapter summarises the main findings and potential directions for future research.

Chapter 2

Background Theory

2.1 Natural Language Processing

Natural Language Processing (NLP) is a subfield of AI that focuses on the interaction between computers and human language. Its goal is to make is to make natural language, unstructured data, understandable and processable by machines through a meaningful numerical representation.

NLP has a broad range of applications across various domains. In customer service and support, it powers chatbots and virtual assistants, enabling them to understand and respond to user queries and requests. In healthcare, it helps in extracting information from medical records, clinical notes, and research papers, aiding in diagnosis, treatment, and drug discovery. In information retrieval, it techniques are used to improve search engines and information retrieval systems, enabling more accurate and relevant results. NLP also plays a crucial role in machine translation, sentiment analysis, text summarization, and many other applications.

In the context of text classification tasks the NLP workflow involves text preprocessing, text representation and classification. However there are different strategies designed for specific use cases and that do also depend on the type of models used: shallow machine learning models and deep neural networks.

2.1.1 Text Pre-processing

A pre-processing step is needed before using unstructured data, as is text, to train any model. Raw text must be converted to a numerical form.

For what concerns ML, text is tokenized, then stop words and noise are removed. Other preprocessing steps, such as stemming or lemmatization, can be optionally performed based on specific requirements or preferences. Tokenization is the process of breaking down a text into smaller units called tokens. Tokens can be words, phrases, or even individual characters, depending on the granularity desired. Tokenization provides a basic structural understanding of the text, allowing for subsequent analysis at a more granular level.

The most common approach to **tokenization** is word tokenization, where the text is split into individual words. However, tokenization can also consider other linguistic units such as n-grams (contiguous sequences of n words). Tokenization can be performed using

simple rules and regular expressions based on spaces, punctuation or contractions. For example, given the sentence "This is the perfect thesis" word tokenization would produce the tokens: ["This", "is", "the", "perfect", "thesis"]. On the other hand, if an n-gram with n=2 (bigrams) tokenization is performed the result would be: ["This is", "is the", "the perfect", "perfect thesis"].

Tokenization faces issues inherent to languages with rich morphology, such as inflected forms and compound words that may require additional preprocessing steps or specialized tokenization approaches. Moreover, the significance of certain punctuation or symbols that carry meaning in the text may be overlooked if tokenization is solely based on white spaces and punctuation marks. It is important to carefully handle punctuation and symbols during tokenization to avoid loss of information.

Stop words removal in NLP involve filtering out common words that do not carry significant meaning in a given context from the text. This process aims to improve the efficiency and accuracy of NLP tasks by reducing the dimensionality of the data and focusing on the more informative content. Stop words are common words that appear frequently in a language but often do not contribute much to the overall meaning of a sentence or text. Examples of stop words include "a," "an," "the," "and," "in," "is," etc. These words are usually grammatical in nature and serve as connectors or functional words.

This is very useful in the ML context to reduce the vector representation of text, nonetheless there might be some disadvantages. For example, in sentiment analysis words like "no" or "not" have contextual importance that determine polarity. Moreover, in domain specific applications a custom stop words list should be considered for frequent words in that field. **Noise removal** involves eliminating irrelevant or unwanted information from the text data. This can include removing special characters, URLs, numbers, punctuation, and other non-textual elements that do not contribute to the primary focus of the analysis. Also for this method excessive data removal could mean loss of relevant information. Therefore, it is important to consider the specific application.

Stemming and lemmatization are techniques used to normalize words, improve generalization, and reduce the complexity of the vocabulary. These techniques aim to group together words with similar meanings. **Stemming** is a process in which the prefixes and suffixes of words are removed, resulting in a truncated or abbreviated form of the word called the "stem." The stem may not always be a valid word, but it represents the core or base form from which related words can be derived. **Lemmatization** is similar to stemming but takes into account the part of speech of the word and ensures that the resulting lemma is a valid word. Lemmatization uses morphological analysis and language resources like dictionaries or WordNet [41] (for the English language) to find the appropriate lemma for a given word.

Nowadays, state-of-the-art results in NLP tasks is obtained by DNN models and in particular by transformers. These architectures are able to process more complex text representations and extract deeper features autonomously. For this reason the tokenization methods such as Byte Pair Encoding (BPE) [55], SentencePiece [30] and WordPiece [53] have gained much interest. The goal is to divide rare words into sub words while keeping together frequently used ones.

BPE operates by iteratively merging the most frequent pair of characters or character sequences until a predefined vocabulary size is reached. The algorithm initializes the vocabulary as the set of all characters present in the corpus. Each character is a token. Then, iteratively merge together the most frequent characters pairs until the iteration limit or desired vocabulary size is reached. BPE has been used in popular NLP models, including OpenAI's GPT-2 [46]. Similar behavior has **WordPiece**, introduced in the context of BERT [13], but prioritizes the merging of pairs where the individual parts are less frequent in the vocabulary with the following score:

$$score = \frac{freq\_of\_pair}{freq\_of\_first\_element \times freq\_of\_second\_element} \quad [1]$$

WordPiece works with a fixed vocabulary size. **SentencePiece** is also a subword oriented method and works by utilizing a unigram language model to determine the optimal subword units for a given text corpus and is designed to handle multilingual and unsegmented text, providing flexibility in tokenizing text into variable-length subwords. It employs an unsupervised learning approach to automatically learn subword units based on the statistical properties of the corpus. Also does not require a predefined vocabulary length.

2.1.2 Text Representation

The preprocessing steps are followed by the creation of a vectorial form of these words lists that can be fed to a model. Text representation indicates those methods that convert textual data into a numerical or computational form that can be processed and analyzed by machine learning algorithms. The most basic one is the **Bag of Words (BoW)** representation through which a text is represented as an unordered set of tokens, and to each token corresponds either the raw value counts or the **tf-idf (term frequency - inverse document frequency)** score which adjusts the raw counts of frequent words [60]. The $tf\text{-}idf$ score of a word t is defined as

$$tf\text{-}idf(t,d) = tf(t,d) \times idf(t) \quad (2.1)$$

where the first term is the number of times a word appears in a given document, while the second is an inverse function of the number of documents in which it occurs.

More advanced representations belong to the family of word embeddings, where each word corresponds to a dense vector that captures its semantic meaning. The **Continuous-bag-of-words (CBoW)** representation obtained through Word2Vec [40] is based on the intuition that a word's meaning can be understood by the surrounding ones. The CBoW model predicts the target word based on its surrounding context words typically using a shallow neural network architecture. A more recent method called GloVe [44], developed by researchers at Stanford University, focuses on capturing global word co-occurrence statistics by constructing a word-context co-occurrence matrix from a large corpus. GloVe then performs matrix factorization techniques, such as singular value decomposition, to obtain the word embeddings.

These kind of embeddings have been used also to train deep learning models such as CNNs and RNNs for text classification [16] to extract more complex relationships. CNNs, usually employed in computer vision, by learning convolutional filters parameters that run over the input text are able to capture a hierarchical representation of the text features. On the other hand, RNNs are networks tailored for processing sequential data, updating

an internal state of the current representation.

However, transformer models have revolutionized text representation in deep learning for text classification as demonstrated in the empirical study of Carvajal et al. [18]. Transformers employ self-attention mechanisms to capture global dependencies and contextual relationships between words. These models learn contextualized representations of text data, enabling a deeper understanding of language and achieving state-of-the-art performance in various NLP tasks.

Transformer embeddings are further strengthened by the pretraining process, where the model is trained on extensive amounts of unlabeled data using self-supervised learning techniques. In the case of BERT [13], the pretraining is achieved through two tasks: Masked Language Modeling and Next Sentence Prediction. Masked Language Modeling involves randomly masking certain words in a sentence and training the model to predict the masked words based on the surrounding context. Next Sentence Prediction task involves training the model to predict whether two consecutive sentences are connected or not.

2.1.3 Text Classification

In text classification the textual representations obtained from embeddings are utilized to train various machine learning models such as SVM, NB, or simple neural networks. When representing documents or sequences of words, a common approach is to average the word embeddings to obtain a fixed-length representation.

In the case of models like BERT, the special [CLS] token is used as a representation for the entire document or sequence. It is important to note that during inference, it is crucial to apply the same preprocessing pipeline and normalization techniques used on the training set to ensure consistency and accurate predictions on new data. This includes tokenization, stop word removal, stemming, or any other preprocessing steps employed during training. Maintaining consistency in preprocessing between training and inference ensures that the input data is processed in the same manner, enabling the model to make accurate predictions based on its learned patterns and representations.

2.2 Transformers

Transformers[65] solve brilliantly many of the issues that limited the use and performance in natural language related tasks of previous sequential and convolutional models. Sequential models, like vanilla RNNs, struggle to capture long-term dependencies due to the vanishing or exploding gradient problem [21]. When processing sequences with long-range dependencies, the influence of early inputs may diminish or explode exponentially over time, leading to a loss of context and affecting the model's ability to understand relationships between distant words. Moreover, the inherent sequential processing of the input hinders the model's capabilities of understanding because it has access only to the preceding words and restricts parallelization capabilities on multiple cores or GPUs.

Long Short Term Memory (LSTM) networks [19] first and **Gated Recurrent Unit (GRU)** networks [10] then have been developed to address the vanishing gradient problem, but the sequential computations with limited scalability and lack of global contextual

understanding remain unsolved problems till the introduction of attention based models.

Transformers have emerged as a revolutionary model architecture in NLP, fundamentally changing the way we approach sequential data modeling. Unlike traditional RNNs that process input sequentially, transformers employ a self-attention mechanism to capture global dependencies and contextual relationships between words or subword units. By leveraging attention mechanisms, transformers can efficiently model long-range dependencies, handle bidirectional context, and parallelize computations, thereby overcoming the limitations of RNNs.

2.2.1 Transformer Architecture

Introduced by Vaswani et al. in the seminal paper "Attention Is All You Need" [65], the transformer architecture fundamentally changes the way we understand and model dependencies in sequences by introducing a self-attention mechanism. This mechanism allows the model to dynamically weigh the importance of different positions within the input sequence, enabling efficient modeling of long-range dependencies and capturing global context. In this section, the details of the transformer architecture are examined, with a focus on its key components, such as the self-attention mechanism, positional encoding, and scalability.

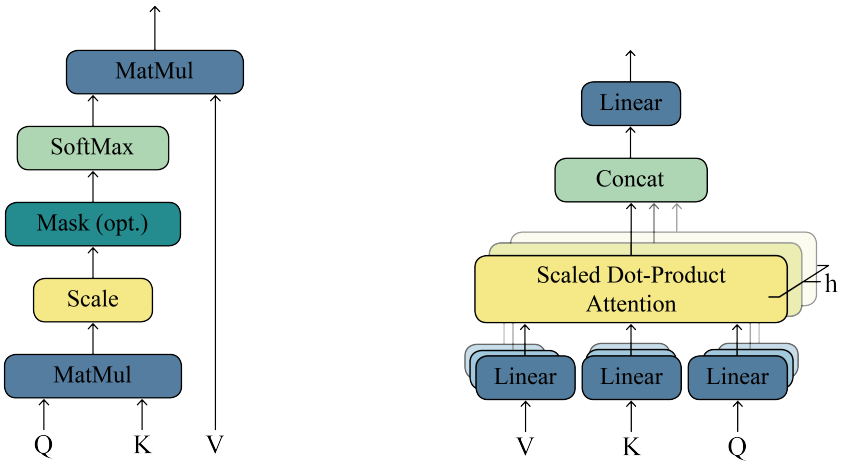
Attention mechanism

The Attention mechanism enables the model to selectively focus on relevant parts of the input sequence while generating an output. The mechanism works by calculating attention weights that determine the importance of each element in the input sequence to the current decoding step. These attention weights are used to compute a weighted sum of the input elements, which serves as the context or representation used by the decoder to generate the output.

The Transformer attention mechanism in particular, also known as **self-attention** or **scaled dot-product attention**, is applied to a sequence of input embeddings. Let $X = [x_1, x_2, ..., x_n]$ be the input sequence, where n is the length of the sequence. For each element in the input sequence X, the **query (Q)**, **key (K)**, and **value (V)** embeddings are derived. These embeddings are obtained by linearly transforming the input embeddings using learnable weight matrices, WQ, WK and WV, respectively. The attention scores are calculated, following 2.1a by first taking the dot product between the query and key embeddings. This dot product is scaled by a factor of the square root of the dimension of the key embeddings dk. The attention scores are then passed through a softmax function to obtain attention weights that sum up to 1.

$$Attention\ score(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.2}$$

The attention weights are utilized to compute a weighted sum of the value embeddings. To capture diverse relationships and enhance performance, the transformer model commonly employs multiple attention heads. Each attention head has its own query, key, and value embeddings, and the outputs of the multiple heads are either concatenated or linearly combined to generate the final representation. 2.1b These dependencies are captured in a parallel, in an efficient manner which makes the transformer model extremely convenient in various natural language processing tasks.



(a) Scaled dot product attention

(b) Parallel execution of attention layers

Figure 2.1: Transformer attention mechanism [65]

Encoder Decoder components

The architecture of transformer models 2.2 comprises an encoder-decoder structure with multiple stacked blocks. The encoder (gray box on the left) takes the input sequence and generates contextual representations. It consists of a set of consequent blocks, which take as input the output of the preceding block, each formed by two sub-layers.

The first sub-layer is a **multi-head self-attention mechanism** that allows each position to attend to other positions in the input sequence, capturing textual dependencies. The second sub-layer is a position-wise feed-forward neural network that provides non-linear transformations to the representations. The **decoder** (on the right) takes the encoder's representations and generates the output sequence. It consists as well of multiple blocks, each with three sub-layers. The first sub-layer is a masked self-attention mechanism that enables the decoder to attend only to previous positions in the output sequence. The second sub-layer is an **encoder-decoder attention mechanism** that allows the decoder to leverage the encoder's representations. The third sub-layer is a position-wise feed-forward neural network as in the encoder ones.
Furthermore, residual connections are used between sub-layers in each encoder and decoder block. These connections allow to pass the original input forward which helps in training deeper networks. Layer normalization is applied after each sub-layer to normalize the outputs and improve training stability. Regarding the input embeddings, they can be learned or pre-initialized with pre-trained word embeddings. Moreover, positional encoding is added to convey positional information.

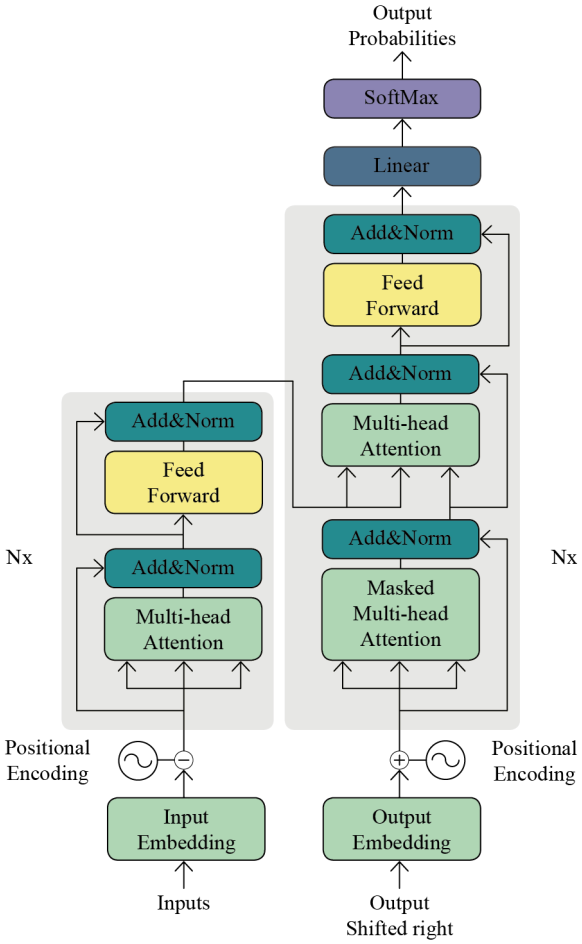


Figure 2.2: The transformer encoder (left) decoder (right) architecture. N stands for multiple stacked encoder/decoder blocks [65]

To the input and output embeddings are added positional embeddings, since there is no implicit encoding of that as happens in RNNs or CNNs. This information is added by summing a vector with the same input dimension d_{model} as the model, for example in the case of BERT $d_{model} = 512$. The positional encoding vector is defined as follows:

$$\mathrm{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right)$$
$$\mathrm{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d}}\right)$$

where pos represents the position within the sequence and i denotes the index of the dimension in the positional encoding vector.

2.2.2 Transfer Learning

Transfer learning involves utilizing knowledge acquired from a source task(s) to enhance the learning process of a target task. The idea behind transfer learning is that models can leverage the learned representations, features, or parameters from a related task to improve generalization and performance on a new, possibly different, task.

Typically, transfer learning consists of two main steps: pre-training and fine-tuning. In

the pre-training phase, a model is trained on a large-scale dataset, often referred to as a source task or source domain. This initial training helps the model learn general features or representations that capture useful patterns from the data.
In the subsequent fine-tuning phase, the pre-trained model is further trained on a smaller dataset specific to the target task or target domain. This fine-tuning process allows the model to adapt and specialize its learned representations to the target task, utilizing the knowledge gained from the source task.

In the case of transformer based language models, pre-training is done through the so called self-supervised training approach. Because no labels are available, an artificial supervised task is created through the large amount of unlabeled examples.

An emblematic example in natural language processing is the **Masked Language Modeling (MLM)** [13] task: words are randomly masked, hidden, or substituted in a sentence and the model is trained to predict the missing word. Another common objective is **Next Sentence Prediction (NSP)** [13], where the model is trained to predict whether two consecutive sentences are coherent or randomly paired. This objective helps the model grasp semantic relationships and capture the coherence of text. Beyond MLM and NSP, there exist several other self-supervised learning objectives, such as **Masked Visual-Language Modeling (MVLM)** [72], image-text alignment[71], or cross-modal tasks. In such an unsupervised manner, the model learns meaningful representations of the data that can be exploited for downstream tasks as classification and information extraction.

2.2.3 Encoder-only Models

Many architectures have been derived from the original encoder-decoder transformer, designed for different tasks.

Decoder-only models are commonly used for tasks like language generation, where the goal is to autoregressively generate coherent and meaningful sequences based on a given context.

Encoder-decoder models are widely used for tasks like machine translation, where the input sequence in one language needs to be converted into a corresponding sequence in another language. The encoder captures the source language information, while the decoder uses that information to generate the target language sequence.

Encoder-only models embed the input sequence into a fixed length vector. Encoder-only transformers have gained significant popularity in various natural language processing (NLP) tasks that do not require generation or decoding, such as text classification, named entity recognition, sentiment analysis, or document classification. These models excel at capturing and encoding the contextual information of the input sequence, enabling them to effectively learn and represent the semantic meaning and relationships between words. In the next paragraphs are illustrated the main encoder architectures and recent developments.

BERT

Bidirectional Encoder Representations from Transformers first introduced by Devlin et al. in [13] is the most widely employed neural architecture for natural language understand-

ing tasks. BERT is an encoder only model, which means it comprises the first half of the layers from the original transformer model [65] responsible for constructing a dense contextual representation of the input sequence. Moreover it exploits bi-directional self-attention which allows each token to attend not only to the tokens on the left but also those that come after in the sentence.

During the pre-training phase self-supervised tasks are employed. The first is NSP, basically a binary classification task: the model is fed two sentences and learns to predict if they are consecutive. The second task is MLM mentioned in the paragraph above. In this way the model learns from massive amount pretrained representations which can then be finetuned according to the task and data at hand.

RoBERTa

Liu et al. understood the potential of pre-training and introduced Robustly Optimized BERT approach (RoBERTa) [34]. As the name suggests it improves the BERT massive pre-training phase. It relies only on MLM but exploits a larger and cleaned pool of data: *BookCorpus* [79] and *English Wikipedia*. RoBERTa also enhances the MLM process with dynamic masking, different masked word at each epoch, showing, in this way, many more combinations to the model. In the following experiments GilBERTo, an Italian pre-trained RoBERTa based [17] transformer is used as embedding model.

Advancements

The transformer architecture has seen several architectural modifications and improvements to enhance its performance and address its limitations. Some notable advancements include sparse attention, performer models, and long-range transformers.

Sparse attention is a modification to the self-attention mechanism in transformers that aims to reduce the computational complexity of attending to all positions in the input sequence. Methods like Linformer [67] introduce sparse patterns in the attention mechanism, allowing the model to attend only to a subset of positions instead of all. This approach reduces memory requirements and speeds up computation while still capturing essential dependencies.

Performer [9] models propose an alternative formulation of self-attention that relies on the Fast Attention Via positive Orthogonal Random features (FAVOR+) algorithm. By using random features, performer models approximate the attention mechanism with linear operations, significantly improving the efficiency of self-attention. This approach offers faster training and inference times while maintaining competitive performance.

Long-range transformers address the challenge of modeling dependencies between distant positions in the input sequence. Standard transformer models struggle with capturing such long-range dependencies due to their self-attention mechanism. To tackle this, models like Longformer [6] introduce mechanisms such as locality-sensitive hashing and sliding window attention, enabling the efficient modeling of long-range dependencies.

2.3 Active Learning

In the realm of machine learning, various approaches have been developed to tackle the challenge of training models with constraints on the amount data. Traditionally, fully supervised learning, in which models are trained using a large amount of labeled data, has been the dominant paradigm. However, as labeling data can be costly and time-consuming, alternative approaches have emerged to leverage unlabeled or sparsely labeled data. Among these approaches, active learning, semi-supervised learning, and few-shot learning have gained significant attention for their ability to achieve competitive performance with the need of fewer labeled examples.

2.3.1 Alternatives to Supervised Learning

Fully supervised learning

Fully supervised learning is the conventional approach, where models are trained using a substantial amount of labeled data. In this paradigm, a large labeled dataset is required to train models effectively. The goal of fully supervised learning is to generalize from the labeled examples and make accurate predictions on unseen data. However, obtaining a large amount of labeled data may be impractical or costly, especially when expert annotations are needed. Indeed, transfer learning offers the advantage of requiring less labeled data compared to fully supervised learning. Nonetheless, it is important to note that even in the case of transfer learning, a considerable amount of labeled data is still required, particularly when dealing with domains that significantly differ from the pre-training dataset or when fine-tuning for downstream tasks large models with millions of parameters.

Semi-supervised learning

In the case of semi-supervised learning, labeled and unlabeled data are combined to train the model. The key idea behind this approach is that the unlabeled data can provide valuable information about the underlying data distribution and improve the model's performance. Semi-supervised learning is based on 3 main assumptions [64]: smoothness, cluster and manifold.

The first assumption allows to propagate labels to near in space points because it states that decision boundaries should be smooth. The cluster assumption follows the smoothness one. In fact, semi supervised algorithms which rely on it can leverage the fact that data consists of clusters and data within each cluster has the same label. Finally, through the manifold assumption data is considered to be near in a lower dimensional space with respect to the original embedding space. By imposing constraints on the model's predictions based on the local relationships between data points, makes easier the job of classifying them.

It is important to keep in mind that these assumptions may not be all true in all scenarios, and the effectiveness of semi-supervised learning techniques can vary based on the specific dataset and task faced. Nevertheless, these assumptions serve as guiding principles in designing algorithms that exploit the benefits of leveraging unlabeled data.

Techniques classified as semi-supervised, differ in the way they exploit unlabeled data and the assumptions they rely on. The self-training technique indicates the process of iteratively using the most confident predictions to pseudo-label data and use it in the next

iteration. A variation of self-training is co-training where different models trained on different features of the same data combine their predictions to pseudo-label the data. Another way of exploiting the unlabeled pool is by conditionally training generative models, which learn the process of creating such data, to build new examples similar to the known ones to be added to the training set.

Few-shot learning

On the other hand, few-shot learning [29] addresses the problem of training models with only a limited number of labeled examples for each class. The goal is to develop models that can quickly adapt to new tasks or classes with limited labeled data by leveraging prior knowledge from a larger dataset or through meta-learning techniques.

- Let $\mathcal{D}_{\text{train}}$ be the training dataset consisting of N classes, where each class C_i contains K support examples $\mathbf{x}_{ij}$ along with their corresponding labels y_{ij}, i.e., $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_{ij}, y_{ij})\}_{i=1,j=1}^{N,K}$.
- Given $\mathcal{D}_{\text{train}}$, the few-shot learning task aims to learn a model $f(\cdot)$ that can effectively generalize to new tasks or classes with limited labeled examples.

Few shot learning can be categorized in 4 main approaches: metric-based, optimization-based and generative approaches. Metric-based approaches, such as meta-learning [32] or prototypical networks [58], learn a representation space such that similar examples are close to each other. At inference time, in the classification case, prediction is made by comparing the new data to known examples in the embedding space and the label of the nearest examples is selected. Methods such as Model-Agnostic Meta-Learning (MAML), Reptile, and First-order Model Optimization (FOMAML) are optimization based [49]. In the case of meta learning the model is trained on a variety of similar tasks and learns to generalize to new tasks by leveraging the learned meta-knowledge. Finally, variational auto encoders (VAEs) [27] are generative methods based on GANs (Generative Adversarial Networks) used to synthesize additional examples that resemble the labeled data, which can then be used to improve the model's generalization.

Active learning

Active learning (AL) is a framework that aims to reduce the labeling effort by selectively querying the most informative instances for labeling. Instead of randomly selecting samples for annotation, active learning algorithms actively seek out the instances that are expected to provide the most valuable information to the model. The underlying assumption is that an algorithm can achieve high performance with less labeled data if it focuses on the most challenging or uncertain examples. The goal of active learning is to achieve performance comparable to fully supervised learning while minimizing the number of labeled instances needed for training.

The core of AL is the query strategy, the function that decides which points from the unlabeled pool need to be queried. According to [62] they can be divided in uncertainty based, diversity based and Hybrid approaches. The first one refers to the uncertainty of the model when inferring the label on unseen examples. In this case the strategy selects the examples on which the model is most uncertain about. Diversity base approaches aim to cover most of the embedding space by selecting examples far from each other. Intuitively, hybrid approaches combine both strategies in order to have an equilibrium between the

two aforementioned criteria.

To summarize, when facing limited annotation budget or a small examples pool, few-shot learning is a promising approach, provided that the performance requirements are not overly stringent. On the other hand, if there is a large unlabeled pool of data available and assumptions can be made about the data distribution, semi-supervised learning shines. While active learning may be the most resource-intensive approach on average, it offers precise control over the labeled dataset and model performance. By selectively querying the most informative instances for labeling, active learning minimizes the labeling effort while potentially achieving similar or even superior performance compared to fully supervised learning. Due to performance requirements Active learning is the focus of the following work and the theoretical background is explored in depth.

2.3.2 AL Scenarios

Active learning can be applied in different application scenarios classified based on how data is made available. What remains constant is the process of querying the oracle with data examples to be labeled. In active learning three main scenarios are possible[62]:

- Membership Query synthesis
- Stream-based sampling
- Pool-based sampling

The concept of **Membership Query Synthesis**[2] pertains to those scenarios where there are limited labeled and unlabeled examples available, and the learning algorithm has the capacity to request the labels for synthetically created examples, which are generated from the given labeled dataset. This means, the algorithm can generate specific instances possessing learnable features, which are then sent to the oracle for labeling. One of the primary challenges in this context is that if the oracle is a human, they might struggle to comprehend and label the synthetic instances, as these might be understood by the model generating them but not by the human operator [5]. It's also worth noting that in many situations, creating useful synthetic data effectively is a complex task. Often, the synthesis of new instances is not required as acquiring unlabeled data is comparatively cost-effective.

This idea led to the development of 'Selective Sampling' strategies [11], designed to choose instances to be sent to the oracle directly from the available pool of unlabeled data.
The first selective sampling category is **Stream-based**[14] active learning, which involves the sequential presentation of examples, one at a time. Stream-based active learning is particularly useful in scenarios with memory and computation constraints. In this setting, the active learning algorithm must make a decision for each incoming example, whether to query it for labeling or not. The challenge lies in balancing the annotation budget effectively. There is a risk of exhausting the annotation budget by selecting less useful examples simply because they were presented earlier, without knowing that more informative examples would come later.

The second selective sampling category is **Pool-based**[31] active learning, where all the unlabeled data is available at once in a pool. Although pool-based active learning can be more computationally expensive, it offers the advantage of being able to evaluate the entire dataset as a whole. This allows for a more comprehensive analysis of the data and

facilitates the effective selection of the most informative examples for annotation. By considering the complete pool of unlabeled data, it enables more informed decisions in sample selection, potentially leading to better model performance.

In this research work, the pool based scenario is considered. In addition to the available datasets, three other vital concepts in active learning need to be considered. First, the choice of the machine learning architecture, whether it is shallow or deep, which influences the suitability of batched or non-batched querying strategies. Another key aspect is the query strategy, which determines how examples are selected at each iteration for annotation. An in-depth exploration of query strategies will be presented in a dedicated section later in this text. Lastly, the stopping criteria, such as query budget, performance requirements or maximum number of iterations, play a crucial role in determining when the active learning process should terminate. Query budget indicates the maximum number of examples that can be labeled through the whole process.

Active Learning pipeline

AL tackles the challenge of training models efficiently when only a limited labeled dataset, $D_L = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, is available, alongside a larger unlabeled dataset, $D_U = \{x_1, \ldots, x_k\}$, where k is significantly greater than n.
The objective of active learning is to iteratively choose the most informative samples from D_U in order to train a model in the most effective manner possible. By strategically selecting the most valuable instances to be annotated, AL aims to optimize the learning process and maximize the model's performance with the available resources.
Active learning techniques can be broadly categorized into two main categories based on how the unlabeled data, D_U, is provided.

In a pool-based active learning approach, the workflow typically follows these steps:

1. Initially, the model is trained using a small, labeled pool, randomly sampled from the available data.

2. The trained model's predictions are then utilized by the query function, which selects the next set of examples to be annotated. This selection process can involve choosing one or more instances that the model finds most uncertain, informative, or challenging.

3. The selected examples are then annotated by human annotators and added to the labeled set, expanding the available labeled data.

4. Finally, the model is retrained using the augmented labeled dataset, incorporating the newly labeled examples.

5. The above steps are repeated iteratively until a predefined stopping criteria is met. This stopping criterion can be based on factors such as reaching a specific performance threshold, exhausting the annotation budget, or achieving satisfactory model performance.

Algorithm 1 formalizes these steps, considering as stopping criteria a limited query budget.

Algorithm 1 Active learning pipeline

Let h_{θ_0} be the model initialized with random weights
Let $D_L(0)$ of size n be a small random sample of labeled data
Train the model using the initial labeled pool: $h_{\theta_1} = \text{Train}(D_L(0), h_{\theta_0})$
Let $S(h, D_U)$ be the sampling strategy
$t = 1$
while budget not exhausted **do**:
    $Q(t) = S(h_{\theta_t}, D_U(t))$ #examples selected to be queried
    $D_L(t) = D_L(t-1) \cup Q(t)$
    $h_{\theta_{t+1}} = \text{Train}(D_L(t), h_{\theta_1})$ #train the model from scratch
    Set $t = t + 1$
return h_{θ_t}

2.3.3 Query Strategies

How the model decides which examples from the unlabeled pool are the most useful to be annotated is at the core of Active learning. In the selective sampling, at each training cycle the model assigns a score to the unlabeled points, according to the employed query strategy, based on which examples are queried and sent to the oracle. According to [62] query strategies can be classified in information based, representation based and meta-active learning approaches.

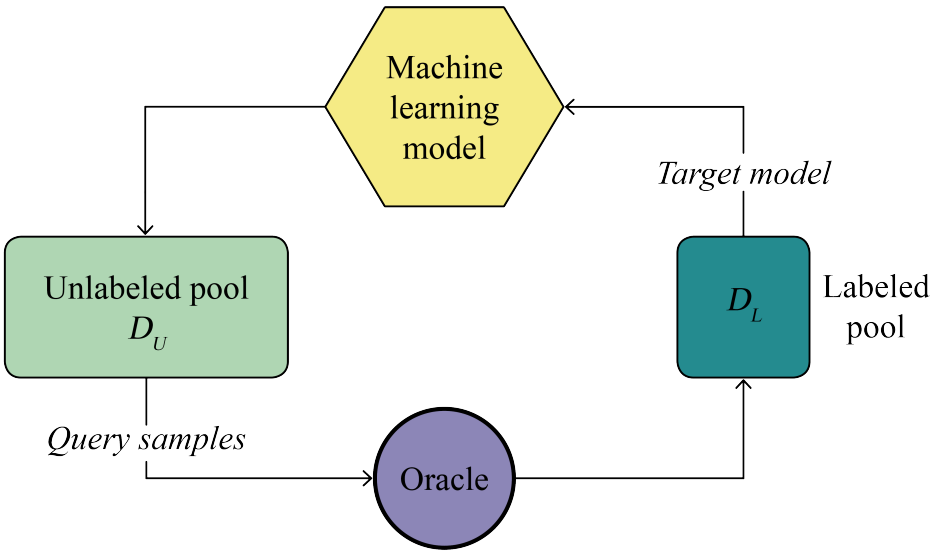


Figure 2.3: Pool based active learning cycle[48]

Information based

First introduced by Lewis et al.[31], information based querying functions, also known as *uncertainty sampling*, such as uncertainty sampling, query by committee and expected prediction change, look out for points where the model is most uncertain about. "Uncertain" points are usually those near decision boundaries, close to each other in the feature space but actually belonging to different classes. There are different versions of uncertainty sampling.
The simplest one queries the instance with **Least Confidence (LC)**, defined as

$$x^* = \arg\max_{x \in D_U} \left(1 - P_h(\hat{y} \mid x)\right). \tag{2.3}$$

$P_h(\hat{y} \mid x)$ is the posterior probability given by model h of the predicted class with highest confidence. The highest confidence class is defined as $\hat{y} = \arg\max_y P_h(y \mid x)$. LC's defect is that it considers only information coming from the most probable label.
Margin Sampling (MS) takes also into account the second most probable predicted label and assigns a score in the following manner

$$MC(x_j) = 1 - P(\hat{y_1}|x_j) - P(\hat{y_2}|x_j) \tag{2.4}$$

where $\hat{y_1}$ and $\hat{y_2}$ are the 2 most probable classes. The score is maximum when the two most probable labels have the same confidence However, it still ignores the distribution of the other labels.
The **Entropy** method, based on the concept introduced by Shannon in [56] addresses this issue and defines uncertainty for each unlabeled example as follows

$$x^* = \arg\max_{x \in D_U} -\sum_i P(y_i|x) \log P_h(y_i|x) \tag{2.5}$$

The summation is done across all possible labels. The entropy is maximized when the all labels are equally probable.
Another category of uncertainty sampling methods is the **Query By Committee (QBC)** approach, in which m models are trained on different feature sets of the data and the examples with the highest disagreement score are queried. The disagreement can be calculated as a form of entropy on the predictions:

$$x^* = \arg\max_x \sum_i \frac{V(y_i)}{m} \log \frac{V(y_i)}{m} \tag{2.6}$$

V is the number of votes obtained by the i-th class.
In the case of **Expected Error (EE)** the goal is to select the example with highest expected future error estimation calculated as:

$$x^*_{0/1} = \arg\max_x \sum_i P_h(y_i|x) \left(\sum_{j=1}^{n_u} 1 - P_{h+\langle x^*, y^* \rangle}(\hat{y}|x^{(j)}) \right) \tag{2.7}$$

For each data point x^* in D_U, the expected future error is estimated by considering the potential labels y^* it could have. This estimation is done using the probabilities P_h assigned by the model to each possible label y_i given the input x. The expected future error can be calculated by summing up the probabilities of all incorrect labels. Here, $P_{h+\langle x^*, y^* \rangle}$

is the new model obtained after retraining it with the labeled data D_L augmented with the selected point $\langle x^*, y^* \rangle$. Select the data point x^* from D_U that maximizes the above expected future error estimation. This means choosing the instance that is expected to reduce the future error of the model the most. Iteratively retraining the model for each example makes this method highly computation expensive and impractical for large datasets.

A more recent uncertainty-based approach is **Bayesian Active Learning by Disagreement (BALD)** [22] which exploits the Monte Carlo approximation method. This method, in the context of Bayesian networks is used to obtain a more reliable confidence measure on the predictions, by running inference multiple times on the data. In the case of neural network's it is done with the dropout technique that allows to "turn off" with a certain probability the network's neurons, typically in the prediction layer. In this way, different versions of the same model can be used to obtain more reliable predictions. BALD is defined as

$$\begin{aligned} I(x_i) = &- \sum_c (\frac{1}{T} \sum_t p(y = c|x_i, \theta_t)) \log_2 (\frac{1}{T} \sum_t p(y = c|x_i, \theta_t)) \\ &+ \frac{1}{T} \sum_t \sum_c p(y = c|x_i, \theta_t) log_2 (p(y = c|x_i, \theta_t)) \end{aligned} \tag{2.8}$$

where the first term is high if the model with different dropouts predicts with high confidence different labels for the same examples or neither version expresses high preference for any of the labels. The second term, the mean of the entropy values calculated for the outputs of the different dropout versions, aims to reduce the score of noisy points which would otherwise be queried.

All these uncertainty-based strategies try to query the examples that the model is most uncertain about. However, there is the concrete risk that when selecting multiple examples per iteration they are similar to each other which is not helpful for an effective model training. Query budget can be exhausted without having explored enough diverse examples.

Representation based

On the other hand, *representation-based* functions aim to use the structure of the unlabeled data to build a subset that best represents the structure of the whole input space. The **density-based** approach retrieves examples from the densest regions of the space, points which are at the minimum distance (using similarity measures) from other points. **Cluster-based** methods select the nearest examples to the centroids of the created clusters.
Finally, **diversity-based** sampling tries to maximize the difference between the examples to query and those already present in the labeled set. One example of a diversity strategy is the **Core-set** [54] approach. In Core-set sampling, data points are selected sequentially, with the goal of choosing the next point that is as far as possible from the previously selected points. The objective is to create a diverse subset by actively seeking out instances that are maximally dissimilar to the ones already chosen. The drawback in this case is that many outliers may be queried and worsen the model's performance. Using exclusively representation-based strategies may require a longer time to reach high classification accuracy.

Meta active learning

The effectiveness of active learning depends on the specific heuristic used to sample from the data. **Reinforcement learning** based approaches formulate the active learning selection problem as a policy to be learned. For example, Woodward et al. [70] used an LSTM deep reinforcement network in a stream based scenario to determine if a data point should be queried or not.

Deep active learning

The techniques discussed in the previous section are highly adaptable and can be applied to any machine learning model, including artificial neural networks. However, there are three main challenges when implementing traditional active learning strategies due to the unique characteristics of complex Deep Learning architectures:

1. Overconfidence: If an Uncertainty Sampling technique is chosen, in neural networks, the prediction vector output from the final layer, which uses the Softmax activation function in a classification context, can be utilized. This vector represents a probability distribution, but neural architectures tend to be overly confident in their class assignment decisions [66], making the obtained values potentially unreliable.

2. Batching: It is unfeasible to train neural networks every time a new labeled example is introduced. Training the network is costly, and a single example barely impacts weight updates. Therefore, for each active learning step where the model is retrained, a substantial batch of new labeled data must be available to add to the training set [77].

3. Inconsistency in the processing pipeline: Traditional active learning strategies focus on training the model for a task and assume fixed input data representations. In contrast, in deep learning, data representations and the task are often learned simultaneously [48].

In essence, deep active learning (DAL) recognizes the distinct requirements of deep neural networks and adapts the selection of examples accordingly. By including larger batches in each iteration of retraining, deep active learning enables efficient and meaningful improvements in the model's performance. By selecting larger batches of examples, deep active learning strikes a balance between training efficiency and the effectiveness of model updates [48].

Kirsch et al. propose **Batch Active Learning by Disagreement (BatchBALD)** [28]. BatchBALD is a strategy that aims to enhance BALD by combining Uncertainty Sampling and Diversity Sampling. Its goal is to select a batch of data points where the model is uncertain while avoiding redundancy in terms of features. The distinctive feature of this strategy is its consideration of the information conveyed by the entire selection process. It takes into account the potential overlap of informative contributions from individual instances within the batch. The metric used becomes:

$$\begin{aligned} a_{batchBALD}(\{x_1, x_2, ..., x_n\}) = &- \sum_{\hat{y}_{1:n}} (\frac{1}{T} \sum_t p(\hat{y}_{1:n}|\theta_t)) \log_2 (\frac{1}{T} \sum_t p(\hat{y}_{1:n}|\theta_t)) \\ &+ \frac{1}{T} \sum_{i=1}^{n} \sum_t \sum_c p(y_i = c|\theta_t) log_2 (p(y_i = c|\theta_t)) \end{aligned} \tag{2.9}$$

BatchBALD calculates the joint distribution expressed by the set of possible outputs from the BNN (Bayesian Neural Network) on the various instances of the batch. If the batch size is large, considering all possible combinations of class assignments $\hat{y}_{1:n}$ becomes infeasible. However, an approximation can still be obtained using the Monte Carlo method. Computing the aforementioned calculation for all possible subsets of size N from the unlabeled pool is also impractical. This is due to the typically extensive collection of unlabeled data. Therefore, the algorithm described in 2 is a greedy approach. It performs N iterations, and in each iteration, the single example that maximizes the value of equation 2.9 on the resulting set is added to the current batch.

Algorithm 2 Greedy BatchBALD

Let N be the query size (number of examples to be annotated)
Let D_U be the set of unlabeled data
$A_0 = \emptyset$ #initialize query set
$n = 1$
while $n \leq N$ **do**
    for each $x \in U \setminus A_{n-1}$ **do**
        $s_x = a_{batchBALD}(A_{n-1} \bigcup \{x\})$
    $x_n = \arg \max_{x \in U \setminus A_{n-1}} s_x$
    $A_n = A_{n-1} \bigcup \{x_n\}$
    $n = n + 1$
return A_n

Batch Active learning by Diverse Gradient Embeddings (BADGE) [3] represents unlabeled examples in a hallucinated gradient space, which effectively embeds both model's uncertainty and data diversity. What distinguishes BADGE from other works that propose hybrid techniques to combine exploitation and exploration is that it does not require in manual hyperparameter tuning. An example of such a hybrid approach is **Wasserstein Adversarial Active Learning (WAAL)** [57]. WAAL utilizes the Wasserstein distance to frame the interactive process of active learning as a distribution matching problem.

2.4 Cold-start

2.4.1 The Cold Start Problem in Deep Active Learning

Definition

The cold start problem was initially identified in recommender systems [76] when algorithms lacked sufficient information about users without any purchase history. Similar issues have surfaced in other fields such as natural language processing and computer vision during the active learning procedure. The Cold Start Problem is an issue that emerges when a machine learning model is tasked to make decisions about the data that it should learn from, while starting from a point of minimal initial training data or sometimes, no training data at all (Hard Cold Start).

In the context of deep active learning, the cold start problem is magnified due to the data-intensive nature of deep learning methodologies. The issue arises when a larger number of annotated instances are required to train a reliable model for the active learning (AL) task. Traditionally, the initial set of samples is randomly selected [78, 20, 51] to

initiate the AL iterative cycle. However, estimating the optimal number of samples needed to train an effective initial model for an AL task can be challenging. Selecting a set of initial samples that is too small or too large can lead to sub-optimal performance given a total annotation budget.

In more formal terms, let $D = \{X, Y\}$ be the dataset, where $X = \{x_1, \ldots, x_n\}$ represents the n data instances and $Y = \{y_1, \ldots, y_n\}$ their corresponding labels. In active learning, the goal is to iteratively select the most informative instances from a pool P (with $P \subseteq X$ and initially, $P = X$) to query their labels and add them to the training set, therefore improving the model's performance with as few labels as possible.

However, at the beginning of this process ($t = 0$), the model has been trained on very few instances (D_0, typically a small randomly selected subset of D), leading to a potentially weak initial performance.

Causes

The cold start problem in deep active learning arises from the fact that an active learning algorithm has to make informed decisions about which unlabelled examples to query for their true labels next, based on the current state of the model. However, at the start, when only minimal or no training data has been provided, the model's initial performance may be too weak to accurately make such informative decisions.
This uncertainty comes from the lack of significant distributional information about the entire data. Consequently, the model cannot effectively estimate which examples would be most informative and beneficial for it to learn from. The main factors that influence the initial poor performance are:

- **Insufficient data**: In the initial stages, the model lacks sufficient data instances to accurately represent the underlying data distribution. Determining the appropriate number of samples needed to create an effective initial model for Active Learning tasks can be challenging. Research has demonstrated that choosing either a too small or too large set of initial samples can result in suboptimal performance, considering a specific annotation budget.[15]

- **Noisy initial phase**: Any initial random choice of instances can lead to poor performance and high variance in the active learning algorithm, which can have long-lasting effects on its learning trajectory. Moreover, selecting data based on the predictions of a model trained on little data may lead to the selection of outliers that will deteriorate also the subsequent iterations' performance.

- **Biased query**: Active learning often exhibits a bias towards specific classes when selecting data. In the initial stages, active querying strategies (2.3.3) struggle to outperform random sampling [8] because certain classes are overlooked during training. This issue arises due to the infrequent occurrence of data from minority classes compared to majority classes. Furthermore, real-world datasets, especially in fields such as medical imaging and the specific business case addressed in this thesis (see figure 3.1), are frequently highly unbalanced. This further amplifies the issue of biased sampling.

Cold start effects and potential risks

The cold start problem can have several detrimental effects and potential risks:

- **Sub-optimal Query Selection**: The model, in its initial stages, may end up querying labels for uninformative instances, leading to sub-optimal use of the labelling budget and poorer overall performance.
- **Increased Computational Cost**: The model may require more iterations and more queried labels to reach a satisfactory performance level, increasing the computational cost and time of the learning process.
- **High Variance in Performance**: The initial randomness and uncertainty can cause high variance in the model's performance, making it harder to achieve stable and reliable results.
- **Inefficiency in Learning**: In the worst cases, the model could fall into a sub-optimal learning trajectory from which it can't recover, leading to permanent inefficiency in learning.
- **Increased annotation efforts**: More AL cycles mean more examples to query, more time spent on annotation, and more annotator cost. It may also be the case where the annotation budget is exceeded in order to reach the desired performance.

2.4.2 Existing Approaches

The unifying theme across existing cold start active learning methodologies is their shared focus on example selection, without leveraging outputs from the target model that's been trained on minimal or no data. A comprehensive review of the approaches found in literature [8, 73, 24, 25, 23] suggests that the primary components of successful cold start sampling strategies are: data representation, data clustering, and representative selection. Furthermore, few methodologies have also been developed that specifically address the challenges posed by unbalanced datasets, such as those proposed by Brangbour et al. [7] and Barata et al. [4].

Here are presented the recent most effective cold start active learning strategies, particularly in the context of pool-based scenarios, are versatile across different domains. Indeed, methods devised for text classification tasks can be easily adapted to image classification scenarios, provided that an appropriate embedding model is employed.

Active Learning by Processing Surprisal (ALPS)

The paper titled "Cold-start Active Learning through Self-supervised Language Modeling" by Yuan et al. [73] introduces a novel approach to active learning (AL) to address challenges faced during the cold-start phase of text classification.

The paper proposes the Active Learning by Processing Surprisal (**ALPS**) algorithm, which uses a pre-trained language model (BERT [13]) to guide data sampling in the cold-start setting. This approach uses the **masked language modeling loss** as a proxy for classification uncertainty. The ALPS algorithm finds examples in the data that are both surprising and substantial, similar to how the highest and most extensive peaks are found in the Alps.
ALPS leverages a self-supervised active learning approach using the language modeling task, which is inherently self-supervised as the label for each token is the token itself. Instead of using the classification loss gradient like previous works such as **BADGE** [3], the authors use the masked language model (MLM) loss to estimate uncertainty.

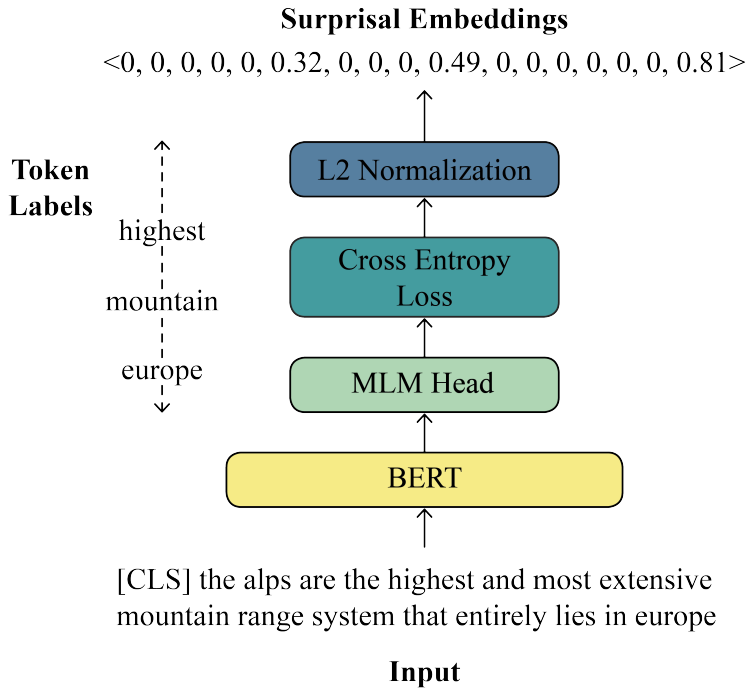


Figure 2.4: The surprisal embedding s_x for sentence x is formed by passing the unmasked sentence through the BERT MLM head, and the cross-entropy loss is computed for a random 15% subsample of tokens against the target labels. The unsampled tokens have zero entries in s_x. The surprisal embeddings are then clustered by ALPS to sample sentences for Active Learning. [73]

To implement ALPS, the authors introduce the concept of **surprisal embeddings**, which are computed by evaluating sentences with the MLM objective (2.4). The embeddings are then clustered using the k-MEANS algorithm to maintain diversity in the selected data. For each cluster center, the sentence with the nearest surprisal embedding is selected for labeling.
The authors demonstrate that ALPS outperforms other active learning baselines (BADGE, entropy and random sampling) in terms of accuracy and algorithmic efficiency when evaluated on four text classification datasets *AG NEWS* [75] (news articles), *IMDB* [36](sentiment reviews), PUBMED [12] (medical abstracts) and SST-2 [59](sentiment reviews). Models fine-tuned with data sampled by ALPS displayed higher test accuracy than the baseline models, and these improvements were observed in the early iterations of the learning process, which is especially crucial in the cold-start setting.

The paper acknowledges that once the cold-start issue is mitigated, traditional uncertainty-based methods could be employed to further optimize the learning process. While ALPS demonstrates superior performance in the cold-start setting, its utility diminishes as more labeled data becomes available and all methods begin to converge in terms of test accuracy.

Cold Start problem in Vision Active Learning (CSVAL)

Chen et al. in their paper "Making Your First Choice: To Address Cold Start Problem in Vision Active Learning" [8] tackle the cold start problem in the field of image classification.

The authors note a discrepancy in the promises of active learning: it often fails to select data as efficiently as random selection in its first few choices. This failure is attributed to a cold start problem, caused by a biased and outlier initial query. The paper specifically addresses this issue in the field of medical image analysis and potentially in the broader field of computer vision.

To mitigate the cold start problem, the authors propose an initial querying strategy that leverages the benefits of contrastive learning: no need for annotation, ensuring label diversity using pseudo-labels, and identifying typical data through contrastive features to reduce outliers.
The methodology involves:

1. **Inter-class Criterion**: To enforce label diversity, the authors employ a K-means clustering algorithm with pseudo-labels and over-clustering to create diverse initial queries. The features required for K-means clustering are derived from contrastive learning methods.

2. **Intra-class Criterion**: To avoid outliers, the method seeks to query hard-to-contrast data. The authors propose a modification of the Dataset Map [26] approach, replacing the ground truth term with a pseudo-label term. This results in querying data that are harder to contrast with others within a cluster, making them more representative of the cluster's distribution.

Experiments conducted on the CIFAR-10-LT and three medical imaging datasets showed that the proposed initial query significantly outperforms existing active querying strategies and random selection.

Additionally, the authors only tested their strategies on academic datasets. In real-world domains data may impose additional constraints on annotation accessibility, annotation costs or annotation confidence. The authors also acknowledged that they focused on standard accuracy and Area Under the ROC Curve (AUC) as evaluation metrics while ignoring other issues in imbalanced data, especially in underrepresented minority classes. These areas present potential directions for future research.

Cold start AL based on Representative sampling (CALR)

Jin et al. in their study titled "Cold-start active learning for image classification" [24] introduce an innovative approach to tackle the cold start problem in vision active learning.

The authors introduce a Cold-start AL model based on Representative sampling (**CALR**), which can select valuable samples without the need for an initial labeled set or iterative feedback from the model. The methodology behind CALR involves three primary components: feature extraction, clustering, and representative selection.

The first component, feature extraction, relies on **contrastive self-supervised learning**

[42]. Because initial labeled samples are unavailable in a cold-start setting, a new approach to learn feature representation is necessary. The authors propose a contrastive self-supervised learning algorithm. This advanced unsupervised feature representation model operates by maximizing the consistency between similar samples (positive sample pairs) and minimizing the consistency between dissimilar samples (negative sample pairs). An encoder network extracts features from the original images and then evaluates the similarity of the sample pairs in the feature space using a similarity score. The encoder is thus able to learn a good representation of images via a "learning comparison" process in the feature space.

The second component is **clustering**, which uses the BIRCH algorithm [74]. BIRCH stands for the Balanced Iterative Reducing and Clustering using Hierarchies, and it is a computationally efficient method that's suitable for large amounts of data and numerous categories. It is a type of bottom-up hierarchical clustering that uses Cluster Features (CFs) to represent a cluster and Cluster Feature Trees (CF-trees) to represent the entire clustering hierarchy. Once a CF-tree is constructed, the clustering results are determined, and the distance between two clusters in each leaf node is calculated. According to the principle of similarity merging, the two nearest clusters are merged into one cluster until the total number of clusters reaches the predefined number of classes.

The third component, **representative selection**, uses a concept called maximum density sampling [39]. The premise is that the sample with the most information should be the one that best represents the potential distribution of a cluster, i.e., the sample located in the densest area of the latent space. The information density of each sample in a cluster is calculated, and the sample with the highest information density from each cluster is selected. This strategy ensures that representative samples can be chosen without needing initially labeled samples or iterative model feedback.

Experimental tests using three image classification datasets (CIFAR-10, CIFAR-100, and Caltech-256) indicated that the CALR model outperforms traditional active learning methods, particularly in low annotation budget scenarios. Furthermore, CALR can be combined with warm-start methods to improve the start-up efficiency and performance of AL.

Despite these advances, the authors acknowledge that their approach has limitations. The effectiveness of their method is particularly influenced by the quality of the feature representation and the unsupervised clustering algorithm, which can greatly influence the success of the initial sampled pool.
Future work could focus on developing more advanced models for cold-start AL that can handle noisy oracle problems. In particular, solutions are needed to mitigate potential loss in accuracy when non-expert oracles are used, due to the potential degradation of annotation quality. Another aspect that requires further investigation is the label ambiguity problem that arises in warm-start AL, particularly when the foreground objects of images are small and the background makes up a large proportion.

2.4.3 Limitations and Research gaps

Experimental Setup and Evaluation

The literature discussed in the previous section primarily conducted experiments comparing cold start techniques to warm start methods within classical active learning cycles,

typically in low-resource settings. These experiments aimed to demonstrate that cold start techniques initially outperformed warm start ones but eventually lagged behind as the active learning process continued. However, a critical limitation arises when applying these findings to real-world scenarios where active learning is employed to optimize performance with fewer labeled examples.

In a practical context, the ultimate goal of active learning is not just to excel in the early stages but to attain peak performance with fewer labeled examples overall. While cold start techniques may provide an initial performance boost, their true value lies in accelerating the subsequent warm start techniques, enabling them to achieve higher performance levels more rapidly. This nuanced perspective on cold start methods and their synergy with warm start techniques is a research gap that demands attention.

In this thesis, the aim is to address this limitation by redefining the evaluation framework. The proposal is assessing cold start techniques not in isolation but in conjunction with warm start strategies. The objective is to investigate how cold start methods can effectively jumpstart the active learning process, facilitating faster attainment of optimal performance in real-world scenarios. By reevaluating and redefining the role of cold start techniques in the active learning pipeline, the goal is to bridge this crucial research gap and enhance the applicability of cold start active learning for text classification of business documents.

Exploitation of Scarce Label Information

In the existing literature, a notable observation is the underutilization of labeled examples, or in some cases, their complete absence in the cold start techniques. While employing a small pool of labeled examples for uncertainty-based methods might seem counterintuitive, it is imperative to recognize that these labeled instances can be leveraged in alternative ways. In studies comparing cold and warm start methods within active learning cycles, the collected labeled examples are often left untapped by cold start techniques. This omission presents a substantial limitation as it signifies a failure to make optimal use of the available information at each cycle.

This oversight is a significant shortcoming of the proposed cold start techniques, potentially resulting in missed opportunities for performance improvement. Effectively exploiting the limited label information at hand could yield a notable difference in performance outcomes. Recognizing this crucial research gap, this thesis endeavors to improve the situation by introducing methods that capitalize on label information throughout the active learning cycles. The goal is to verify if harnessing even a small pool of labeled data can lead to substantial performance enhancements, thereby pushing the boundaries of cold start active learning for text classification of business documents.

Validation of Results on Real-World Datasets and Different Domains

Finally, methods proposed in the literature are tested only on academic datasets which do not present the difficulties of real-world datasets such as noise, etc. This omission is a notable limitation as it fails to account for the complexities and challenges encountered in practical applications. The passive focus on academic datasets potentially overlooks critical nuances that are prevalent in real-world scenarios. Addressing this research gap is imperative, as it is crucial to validate the efficacy and robustness of cold start active learn-

ing techniques in authentic, diverse datasets representing various industries and document types.

By directing attention to these three fundamental areas - experimental setup and evaluation, exploitation of scarce label information, and validation on real-world datasets and different domains - this thesis' goal is to contribute to the evolution of cold start active learning for text classification of business documents, providing a more comprehensive and applicable framework for real-world applications.

Chapter 3

Methodology

This thesis has been structured as an experimental study, thus experiments are conducted to verify hypotheses and validate assumptions. These hypotheses and the design of the experiments have been influenced by extensive reviews of existing literature on cold start problems, ensuring the work is grounded in theory and continues the line of investigation from previous research.

The research design involves not just the focus on outcomes, but a comprehensive investigation of the process as well. Detailed visualizations and in-depth analyses of the intermediate steps during the experimentation process are included, allowing for a thorough understanding of the advantages and disadvantages of each implemented approach.

A significant contribution of this work to the field is the introduction of novel approaches to cold start active learning in text classification. These new methods and techniques, developed as extensions of current knowledge in the field, are subjected to a robust comparative analysis with established techniques, providing an assessment of their relative performance and feasibility.

Unlike many existing studies, the experiments conducted as part of this work make use of real-world datasets. This allows for a more authentic application scenario to be presented, introducing complexities and challenges not typically encountered with the sanitized datasets often used in academic publications. The use of these real-world datasets not only tests the robustness of the methodologies and proposed approaches under conditions that closely mirror real-world scenarios, but also offers insights that are directly applicable and transferable to practical business settings.

3.1 Dataset

A client of Altilia, operating in the financial sector, has provided a dataset consisting of legal documents intended for classification and subsequent information extraction. The dataset includes nine distinct classes of documents. Eight of these classes represent different document categories, each of which will be subjected to key information extraction post-classification. The ninth class, termed *altro* comprises all documents that do not fall into any of the preceding categories. The *altro* class essentially encapsulates documents lacking any actionable information.

The dataset itself is a collection of multi-page scanned documents. The client has as-

signed each document to one of the nine classes. It is worth noting that not every page within a given document necessarily pertains to the assigned document category. To address this, further annotations were made in-house, categorizing each individual page into one of the nine mutually exclusive classes. Consequently, each multi-page document could potentially be segmented into various subsets of continuous pages, each subset adhering to a particular document type. Thus, a single document may comprise multiple sub-documents.

To transform the scanned documents into a workable format, an off-the-shelf OCR (Optical Character Recognition) tool was employed: TESSERACT [43]. This tool was used to extract raw text from each scanned page. Consequently, each page in the dataset is represented by its text content, while the associated image content will be reserved for the subsequent key information extraction task, which is beyond the scope of this study.

The class distribution of the pages is illustrated in figure 3.1 and reveals an imbalance within the dataset. The classes *altro* and *contratto di mutuo* are the most prevalent, whereas *ordinanza di vendita* is the least represented. Further analysis, as depicted in the Appendix (see A.1.1), shows that most documents consist of a mixture of pages, with a significant portion falling into the "altro" category. Additionally, each page contains approximately 300 tokens on average, with some pages containing upwards of 1000 tokens.

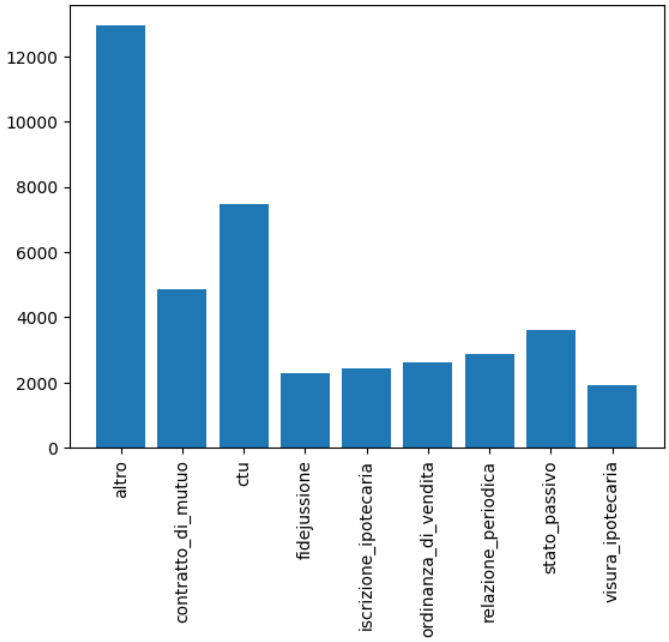


Figure 3.1: Dataset class distribution

For the purposes of this study, we retained the original document structure, allowing for the presence of non-homogeneous multi-page documents. Despite our focus being on page classification, the split between training and development sets was implemented at the document level, ensuring all pages of a particular document remained within the same set. This approach was taken to prevent the possibility of pages from the same document appearing across sets, which could compromise the model's performance validation.

For a more rigorous evaluation of the trained model, we conducted a 5-fold cross-validation, allocating 80% of the data for training and 20% for development. Despite the document-

level split, the distribution of page classes within the training and development sets closely mirrored that of the original dataset. Figure A.1.2 in the appendix displays the distribution of page classes within the five training sets, following the removal of any empty pages.

3.2 Baseline models

Preliminary work has been carried out before the actual focus of the research on the cold start problem. First, it was evaluated the difficulty of the page classification on the whole dataset. After that, different active learning were compared.

3.2.1 Page Classification

Problem definition

In literature, by document understanding refers to tasks related to single-page documents. In this report, to avoid ambiguity, document is used to refer to a multi-page set while page to refer to the single example. In order to precisely classify documents composed of multiple sub documents, the approach taken is classification of each page of a document. In this way, the document is segmented and categorized at the same time. Therefore, given a set of document pages' text, a DL model is trained to assign one out of nine labels to a new unseen page, a multi-class classification problem.

In mathematical terms, the task of multi-class page classification can be defined as follows. The set of pages is denoted as $P = \{p_1, p_2, ..., p_n\}$, where each page p_i represents a piece of text. Additionally, a set of pre-defined classes $C = \{c_1, c_2, ..., c_k\}$ is assumed, where each class c_i represents a distinct category that a page can belong to. Then, the labelled dataset can be seen as a set $L = \{(p_1, c_1), (p_2, c_2), ..., (p_n, c_n)\}$, where the tuple (p_i, c_i) represents a page-label pair.
The goal is to train a model that can automatically assign the appropriate class label to each page based on its content. This classification task can be represented as a function $f : P \mapsto C$, where f takes a page p_i as input and outputs the predicted class label c_i.

Experimental setting

In this experiment, the transformer model trained to embed and classify pages is GilBERTo [17]. GilBERTo is a specialized model based on the RoBERTa architecture [34] that has been pre-trained specifically for the Italian language. The maximum input token length for RoBERTa-like models, including GilBERTo, is 512 tokens. Therefore, inputs longer than that have been truncated. The architecture of GilBERTo comprises a stack of 12 transformer encoder layers. The hidden states, the internal vectorial representation of text, are of dimension 768. To adapt the pre-trained GilBERTo model for page the current task, a classification head is added on top of the encoding layers. The classification head comprises a linear layer that maps the pooled high-dimensional hidden states from the encoding layers to the log probabilities (logits) of the 9 output classes. The model is trained with cross entropy loss on the softmaxed logits.

The GilBERTo tokenizer follows the approach of the CamemBERT [38] tokenizer which is based on an implementation of Byte-Pair Encoding (BPE) called SentencePiece [30]. During the tokenization process, the input text is first segmented into individual words and subwords using SentencePiece. The tokenizer then maps each word or subword to a

corresponding token. Special tokens, such as the start-of-sentence (<s>), end-of-sentence (</s>), and padding (<pad>) tokens to match the input sequences length, are also added. If a sequence exceeds the maximum input length it is truncated.

The GilBERTo model is trained for 5 epochs using a learning rate of 5×10^{-6}. A batch size of 16 is utilized to efficiently process the training data. The AdamW [35] optimizer, known for its adaptive learning rate and weight decay regularization, is chosen for training the model. The model and tokenizer utilized in the experiments were sourced from Hugging Face's [69] library.

Metrics

In evaluating the performance of the trained model, several metrics are employed to comprehensively assess its effectiveness in page classification. The following metrics are utilized:

- Overall Weighted-F1
  - The weighted-F1 score is a weighted average of the F1 scores for each class i, considering their support.
  - This metric provides an overall assessment of the model's performance, taking into account both precision and recall for all classes.

$$weighted\text{-}F1 = \frac{\sum_{c_i \in C} (F1(c_i) \times support(c_i))}{\sum_{c_i \in C} support(c_i)}$$

- Macro-F1
  - The macro-F1 score calculates the average F1 score across all classes, giving equal importance to each class.
  - It provides insights into the model's performance in terms of precision and recall, irrespective of class imbalance.

$$macro\text{-}F1 = \frac{\sum_{c_i \in C} F1(c_i)}{|C|}$$

- Micro-F1
  - The micro-F1 score computes the F1 score by considering the overall true positives, false positives, and false negatives across all classes.
  - It treats the classification task as a single multi class problem and provides a performance measure for the overall classification accuracy.

$$micro\text{-}F1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

- Accuracy
  - Accuracy represents the proportion of correctly classified samples out of the total number of samples in the dataset.

- It provides a simple and intuitive measure of the model's overall classification performance.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

For each individual class, the following metrics were calculated:

- Precision
  - Precision quantifies the proportion of correctly predicted positive samples (true positives) out of all samples predicted as positive (true positives + false positives).
  - It assesses the model's ability to avoid false positive predictions.

$$precision(c_i) = \frac{TP}{TP + FP}$$

- Recall
  - Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive samples (true positives) out of all actual positive samples (true positives + false negatives).
  - It evaluates the model's ability to identify all positive instances.

$$recall(c_i) = \frac{TP}{TP + FN}$$

- F1
  - The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance.
  - It combines both precision and recall into a single metric and is useful for evaluating the model's effectiveness in classification tasks.

$$F1(c_i) = \frac{2 \times precision(c_i) \times recall(c_i)}{precision(c_i) + recall(c_i)}$$

Evaluation

Category	Precision		Recall		F1	
	avg (%)	std (%)	avg (%)	std (%)	avg (%)	std (%)
Altro	92.7	2.0	85.8	1.4	89.1	1.6
Contratto di Mutuo	90.9	2.6	93.6	2.8	92.2	1.2
CTU	90.0	3.3	95.9	1.3	92.8	1.8
Fidejussione	97.2	0.8	97.2	2.1	97.2	1.1
Iscrizione Ipotecaria	97.2	2.2	98.8	1.4	97.9	1.1
Ordinanza di Vendita	88.5	4.2	92.5	1.7	90.4	1.5
Relazione Periodica	89.3	4.8	95.4	1.7	92.2	2.4
Stato Passivo	96.2	3.0	97.7	0.9	96.9	1.8
Visura Ipotecaria	95.7	2.4	91.6	7.3	93.4	3.9

Table 3.1: Average class metrics over 5-folds

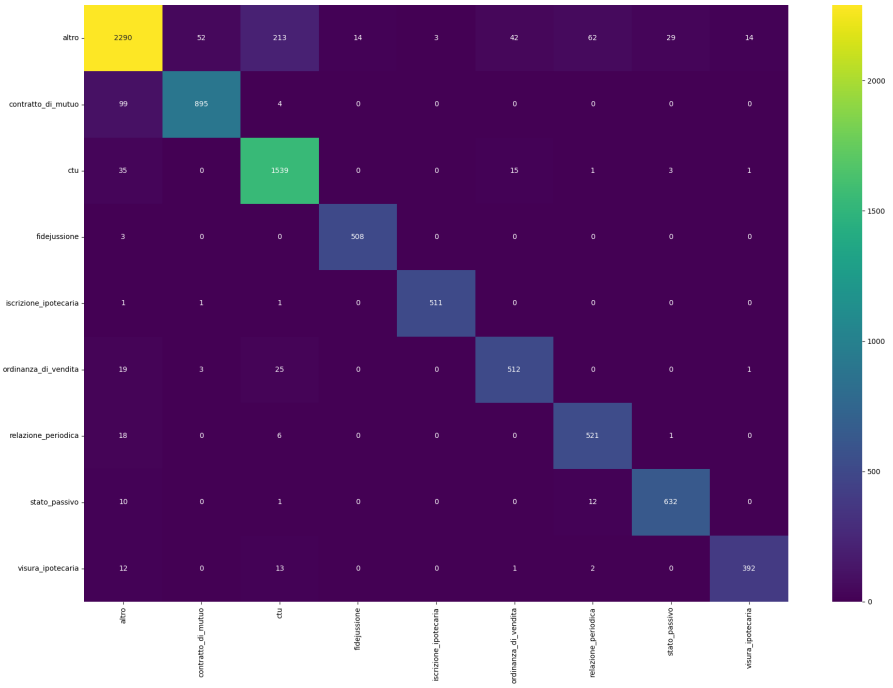


Figure 3.2: Confusion matrix of first fold

Results are shown in the two figures above. Table 3.1 presenting the average metrics for the nine classes across five folds, while the second figure 3.2 is a confusion matrix related to the first fold.

From the table, it is evident that each class achieved an F1 score of over 89%, indicating a high level of classification effectiveness. However, the class *altro* exhibited the lowest performance after *ordinanza di vendita*. Specifically, the model demonstrated a relatively low recall of 85% for the *altro* class, implying that it misclassified some of the *altro* pages with another label. Moreover, it mistakenly classified as *ordinanza di vendita* other types of pages, leading to an 88% precision score for that class.

The confusion matrix helps visualize the challenging areas where the model struggles to differentiate between two classes. In particular, the presence of 429 misclassified *altro* pages contributes to the low recall. Furthermore, the majority of misclassifications occurred between *ctu*, *ordinanza di vendita*, and *relazione periodica*, which explains the low precision for these classes.

The weighted average F1 score yielded an impressive value of 92.4%. These observations highlight the specific areas where the model may require further improvement, such as better distinguishing between the *altro* class and correctly classifying pages related to *ordinanza di vendita*.

Discussion

The company's performance requirement for this task is an average weighted f1-score of 85%. This model exceeds the requirement by 7 percentage points. This result shows an overall marked separability between classes. Moreover, quiet unexpectedly also the class altro obtains an average F1 score above 90%, an average precision of 93% and recall of 86%, which could mean that even if it is an implicit class that collects any kind of document excluding the other 8 classes, its documents are actually different from the other classes and their representations in the embedding space are meaningful. In other words, there are not so many different kinds of documents included in the dataset labeled as *altro*. With careful inspection of the dataset, it is noticeable that the altro documents are usually attachments to other multi-page documents containing emails, copies of other documents, unofficial pre-prints.

It is worth noting that attaining these results required considerable effort and time from Altilia's annotators, who meticulously annotated a dataset of 30,000 examples. However, there remains a question as to whether similar results, or at least results meeting the requirements, could be obtained with a smaller training dataset. To investigate this, 5 random subsamples were extracted from the original dataset, while keeping the development set unchanged. The subsamples consisted of 50%, 25%, 12.5%, 6.25% and 1% of the original training data. By examining the model's performance on these subsamples, insights can be gained regarding the minimum training data required to achieve satisfactory results. The results of these experiments are shown in the table 3.2 below.

Training Set Ratio	Average Weighted F1	Standard Deviation
100%	92.4%	1.12%
50%	90.45%	1.43%
25%	89.16%	1.34%
12.5%	86.21%	1.58%
6.25%	80.00%	0.80%
1%	14.62%	1.82%

Table 3.2: average Weighted F1 score on random subsamples

The findings of this study demonstrate that achieving the desired performance requirements in document classification can be accomplished using only 12.5% of the training data, which corresponds to around 3'750 pages with respect to 30'000 pages. These results provide a strong motivation for further research in developing methods that enable effective and efficient gradual training of models.

3.2.2 Active Learning

Problem definition

To formalize this pool-based active learning experiment, let $D_L = \{(x_j, y_j)\}_{j=1}^{M}$ be an initial labeled set, and a large pool of unlabeled data, $D_U = \{x_i\}_{i=1}^{N}$, where $M << N$. The class label of x_i is denoted as $yi \in \{1, ..., k\}$ for multi-class classification. In each

iteration, a batch of samples, D_Q, with a batch size of b is selected from Du based on the learned model M and an acquisition function denoted as $\alpha(x, M)$. The labels of the selected samples are queried from an oracle, in this case simulated. The samples are chosen using $D_q^* = \arg\max_{x \in D_U}^{b} \alpha(x, M)$, where the superscript b indicates the selection of the top b points. The labeled set, D_L, and the unlabeled pool, D_U, are then updated, and the basic learned model is retrained using D_L.

Experimental setting

In these paragraphs it will be described the practical setting of the conducted experiments. The techniques used are entropy-based sampling, coreset, coreset plus entropy (corentropy) and BALD. Random sampling is used as a baseline. Entropy's implementation follows precisely what described in the related work section 2.3.3. A greedy approach described in algorithm 3 is used to implement the coreset approach where examples are selected one by one.

Algorithm 3 Greedy Core-set

1: Let D_U be the set of unlabeled data
2: Let D_L be the set of labeled data
3: Let Δ an arbitrary distance measure
4: Let N be the query size (number of examples to be annotated)
5: $s = \emptyset$ #initialize query set
6: **repeat**
7: $\quad u = \arg\max_{i \in D_U \setminus s} \min_{j \in D_L \bigcup s} \Delta(x_i, x_j)$ #x_i and x_j are the examples' feature vectors
8: $\quad s = s \bigcup u$
9: **until** $|s| \neq b$
10: **return** s

At step 7, it is selected the unlabeled example which has the maximum distance from its nearest example within the partial solution set. Corentropy's objective is to balance between uncertainty and diversity sampling by linearly combining entropy and diversity measure as follows:

$$Corentropy(x_j) = \lambda CS(x_j) + (1 - \lambda)H(x_j) \tag{3.1}$$

In equation 3.1 the coreset strategy (CS) is called over the unlabeled pool with a budget of size as the length of the unlabeled pool to have a distance measure for each data point. Entropy (H) values are comprised between 0 and 1, so the distances returned by coreset are normalized consequently.

For what concerns BALD, the number of iterations for which the model is run with different dropouts on the same data is set to 5.

While there are numerous libraries offering pre-implemented active learning techniques like DeepAL[1], modAL[2], Distil[3] and baal[4], they have been manually implemented to en-

[1] https://github.com/ej0cl6/deep-active-learning.git
[2] https://github.com/modAL-python/modAL.git
[3] https://github.com/decile-team/distil.git
[4] https://github.com/baal-org/baal.git

sure complete customization and integration with recent transformer models and also to incorporate the corentropy strategy effectively.

Active learning iterations

An initial, randomly sampled, set of 400 pages is used to train the model for the first iteration. Afterwards, the selection budget is set to 1600 examples per iteration for 20 iterations in total. The classification model is evaluated with the same metrics used in the page classification experiment 3.2.1. However, the quality of the active learning strategy does not depend only on the last iteration performance, thus, it is important also to look at the area under the curve across the iterations.

Evaluation

In figure 3.3 is shown the weighted F1 score of the model on the development set at each of the 20 iterations. As mentioned before, the BALD sampling technique was interrupted at the 9th iteration due to its high computational cost and low performance. Around the 10th iteration performance peak (over 90% weighted-F1) is reached by all techniques except BALD. Random sampling shows a comparable performance to the best performing techniques and larger area under the curve. It can also be noticed a large weighted-f1 score jump from iteration 0 (0.15) and iteration 2 (85%, 80%, 73%, 68% for random, entropy, corentropy and BALD respectively) Overall BALD sampling is by far the worst performing technique.

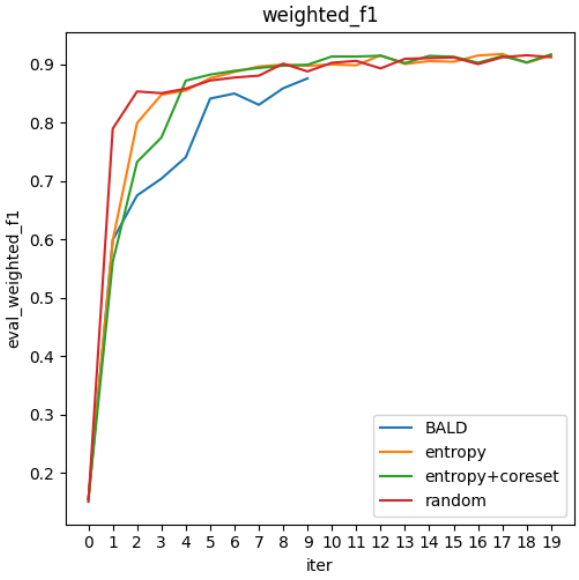


Figure 3.3: Weighted F1 across 20 AL cycles

Discussion

Despite its simplicity entropy has been shown to be an effective active learning technique [52] but these results apparently do not confirm this. The reason this happened could be just the effect of the specific seed selected for random sampling or the model is not effectively capable of expressing correctly confidence on its predictions. It is also possible that entropy is actually not enough in this case to beat a random sampling approach. One way to reduce results variance and have more significant results is to perform k-fold

cross validation, and repeat the experiment with different seeds for randomly initialized strategies. Furthermore, as the performance delta (65% for random sampling) shows in the first 2 iterations it would be useful to reduce the budget per iteration in order to have a more fine-grained perception of how the sampling technique is doing. It might also be the case that querying batches of size 1600 nullifies the capabilities of active learning strategies.

The techniques used in this experiment select examples one by one which is not ideal in the case of batch-based settings because the selected examples could be similar to each other and not exploit effectively the query budget. Other batch oriented strategies may obtain better results in this setting. Further experimentation has to be carried out with more advanced techniques such as BADGE which combines uncertainty and representativeness along with a batch approach.

3.3 Cold start strategies

This section details the implementation of various cold start active learning techniques, as they are employed in this thesis. A selection of established approaches drawn from the existing body of literature, as well as new techniques proposed as part of this work, are discussed and compared. The comparative analysis allows for an understanding of the relative performance and viability of the traditional methods in contrast with the innovative methods introduced in this study.

These techniques are evaluated within the framework of their ability to address the complexities and challenges inherent in cold start active learning for text classification. As discussed earlier, they will be evaluated based on their effectiveness in providing a better initial labeled pool to kick-start warm start AL strategies. The forthcoming discussion aims to illuminate the particular characteristics, advantages, and potential limitations of each approach as they are applied in a real-world context. The methods proposed in this thesis leverage principles of unsupervised learning and incorporate the latest advancements in transfer learning and transformer architectures.

These sophisticated techniques, some of which extend beyond the scope of the prior literature review, will be further elucidated within this section. Each technique's explanation will encompass an exploration of its underlying concepts, providing a comprehensive understanding of how they contribute to addressing the complexities of cold start active learning for text classification. Some of the proposed approaches in this work try to exploit the little knowledge that is made available by querying a small number of samples that would not be enough to perform the classic fine-tuning and uncertainty measure extraction needed for active learning techniques. As identified by numerous ablation studies in the work by Jin et al.[24] important features that make an initial set good are:

- **Level of label diversity**: the balance in the representation of each class within the sample. An ideal initial pool should not be skewed towards a particular class but should instead have a balanced representation from all classes. This balance ensures that the learning algorithm has a sufficient and varied set of instances from each class to learn from, which aids in creating a more generalized and robust model.
- **Inclusion of typical data**: the selection of representative instances from each class while minimizing the inclusion of outliers. Representative or typical data provide the most informative examples of each class, and hence, are crucial for model training.

At the same time, the exclusion or minimization of outliers in the initial pool helps prevent the model from being misled during the learning process. Outliers could cause a disproportionate influence on the model, leading to overfitting and thus a poorer generalized performance.

In sum, label diversity ensures an even distribution across all classes, while the presence of typical data assures the selection of the most representative instances from each class. Both these factors combined provide a solid foundation for effective active learning.

3.3.1 T-CALR

Inspired by the study "Cold-start active learning for image classification" by Jin et al.[24], the proposed approach in this thesis is T-CALR (Textual Cold-start Active Learning model based on Representative sampling). T-CALR addresses the dual challenge of selecting a diverse set of examples from different labels and choosing typical, or representative, data.

Jin et al.'s original CALR methodology tackled the diversity problem by clustering image embeddings into a number of clusters equal to the number of classes. CALR's clustering is performed using the BIRCH algorithm (discussed in detail in section 2.4.2). This approach tried to ensure a balanced representation of examples from each class. For the selection of typical data within each cluster, CALR leverages an **information density** score. Each example within a cluster is assigned this score, calculated as follows:

$$\text{InformationDensity}(x_i) = \frac{1}{n} \sum_{j}^{n} sim(x_i, x_j) \tag{3.2}$$

In this formula, n represents the number of examples within the considered cluster. Essentially, points that demonstrate greater similarity to other points within the cluster are assigned a higher score. This similarity, represented by $sim()$, is evaluated using **cosine similarity**:

$$\text{CosineSimilarity}(x_1, x_2) = \frac{x_1 \cdot x_2}{|x_1||x_2|} \tag{3.3}$$

Although CALR was originally designed for image classification tasks, exploiting a contrastive learning framework for image feature extraction, it can be adapted effectively for textual data. In this thesis, T-CALR employs representations derived from transformer models, specifically Sentence BERT transformers (SBERT), to facilitate the application of the CALR approach to text classification tasks. The three main steps of the T-CALR algorithm, sentence embedding extraction, hierarchical agglomerative clustering and representative selection, are shown in figure 3.4.

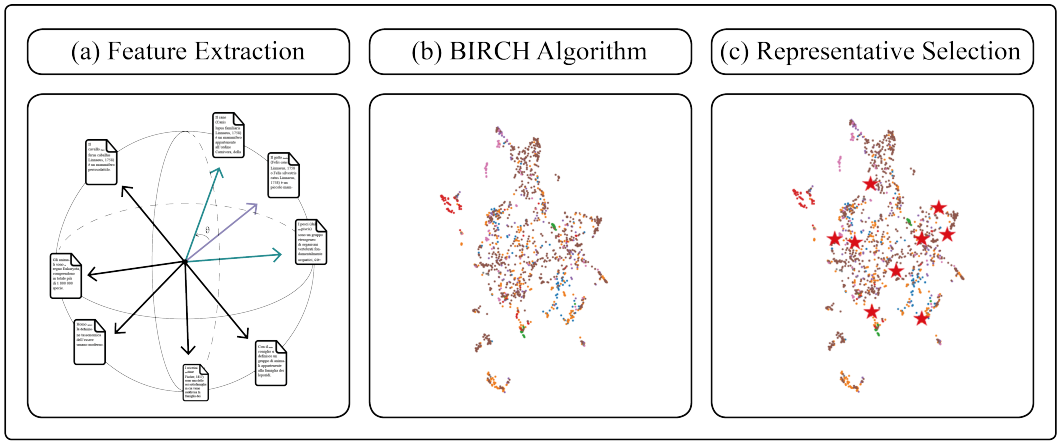


Figure 3.4: The T-CALR approach can be summarized in three consecutive steps: a) feature extraction with SBERT, b) clustering and c) Representative sampling with information density ranking

Sentence Bert

Sentence Transformers, also known as Sentence-BERT (SBERT) [47], are a modification of the BERT [13] architecture that has been specifically optimized for sentence-level tasks. Developed by researchers at the UKPLab, Sentence Transformers offer an effective solution for various NLP tasks that require sentence embeddings.

The architecture of Sentence Transformers is essentially a Siamese or twin network structure. This design comprises two identical neural networks (transformer models), each taking one sentence as input. Both networks share the same parameters, implying that they are 'twins.' The output is then the vector representations of the input sentences, which can be directly compared to compute semantic similarity.

During the training process, Sentence Transformers are taught to produce sentence embeddings that are semantically meaningful. This is achieved by using various types of training data, including parallel corpora (such as translated sentences) or pairs of sentences from tasks like Natural Language Inference (NLI).

Sentence-BERT (SBERT) introduces a pooling operation to the output of BERT or RoBERTa models, creating fixed-sized sentence embeddings. The MEAN pooling strategy is employed as the default configuration. BERT models are fine-tuned through the creation of siamese and triplet networks. This adaptation facilitates the production of semantically meaningful sentence embeddings, which can be compared using **cosine similarity**. The specific

network structure varies depending on the available training data, and different structures and objective functions have been experimented with: classification objective function, regression objective function and triplet objective function.

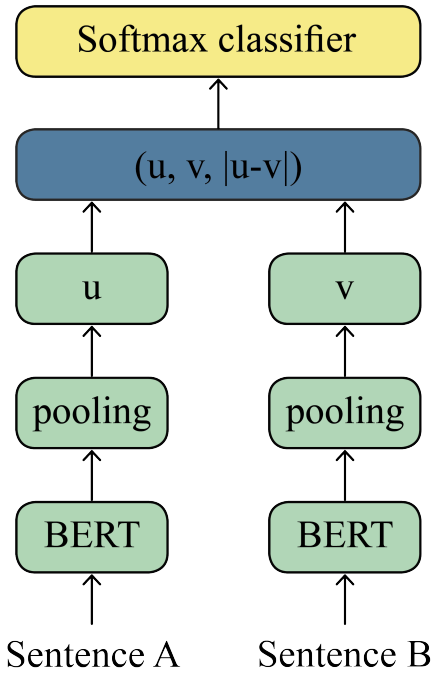


Figure 3.5: Siamese BERT architecture with tied weights, classification objective function

Sentence Transformers present an efficient way to compute sentence embeddings, enabling better performance and faster computation times for many NLP tasks that require understanding the semantic similarity between sentences. In the paper [47] it is also shown the superior effectiveness of Sentence Transformers in comparison to traditional BERT embeddings, particularly in unsupervised tasks.

The Sentence Transformer model is specifically fine-tuned to generate sentence embeddings that capture semantic information at the sentence level, resulting in more meaningful and contextually relevant representations. This contrast markedly with traditional BERT embeddings, which are primarily designed for word-level tasks.

Consequently, Sentence Transformers often outperform BERT embeddings in unsupervised tasks that require sentence-level understanding, such as semantic textual similarity, clustering, or information retrieval. Their ability to capture the comprehensive semantic information of a sentence in a single vector allows for more accurate comparisons between sentences and more meaningful groupings in unsupervised clustering tasks.

Strengths and weaknesses

T-CALR is an unsupervised approach for a representative selection of an initial labeled pool to start the active learning process. It exploits pre-trained embeddings provided by the SBERT model and unsupervised agglomerative clustering BIRCH. T-CALR does not rely on any feedback from the data, neither uncertainty measures nor labels from the annotators.

This makes it an efficient sampling strategy because it can select in one shot all the needed examples for the initial pool. It addresses label diversity with clustering and typical data selection with information density scores but it has some drawbacks.

First, being completely unsupervised its effectiveness is closely related to the effectiveness of the BIRCH in arranging the data points. Second, when selecting the most information dense examples from each cluster there is the concrete risk of selecting examples similar between each other. Selecting examples close in the embedding space of a cluster may not be the best way to represent it and effectively use the available budget.

3.3.2 Iterative T-CALR

Building upon the foundation laid by T-CALR (Textual Coldstart Active learning with Representative sampling), an upgraded strategy named Iterative T-CALR is introduced. T-CALR is a three-step process that includes feature extraction using SBERT, clustering via the BIRCH algorithm, and representative selection using information density.

Iterative T-CALR, while retaining the basic framework of T-CALR, introduces significant enhancements inspired by the few-shot learning framework, SetFit [63]. SetFit is renowned for its ability to fine-tune models with minimal data, making it an ideal fit for cold start active learning scenarios, where limited labeled data is a defining characteristic. SetFit is a few-shot learning approach that bridges the gap between traditional machine learning and human-like learning, which requires only a few examples to learn new tasks. Its method includes creating a set of feature vectors and optimizing a learnable transformation to align it with the ground-truth feature set. In doing so, it can fine-tune models with limited data effectively.

Iterative T-CALR diverges from the original T-CALR process at the point of feature extraction. Instead of using a fixed SBERT model for this task, Iterative T-CALR utilizes the SetFit framework to fine-tune the SBERT model at each iteration. The resultant, fine-tuned SBERT model is then used to extract features.
As a result, the subsequent clustering and representative sampling steps operate on fresh and potentially superior embeddings provided by the continuously trained SBERT model. This iterative process, therefore, allows for a continuous refinement of the model and the embeddings it generates, thereby enhancing the performance of the active learning process over time.

SetFit

SetFit[63] (Sentence Transformer Fine-tuning) is an efficient and prompt-free framework for few-shot fine-tuning of Sentence Transformers (ST). It dispenses with the need for prompts and does not require large-scale pre-trained language models (PLMs) to achieve high accuracy, even with a small number of labeled examples. Inspired by Sentence Transformers

[47] introduced earlier, which utilize Siamese and triplet network structures, SetFit aims to derive semantically meaningful sentence embeddings. Its primary goal is to minimize the distance between pairs of semantically similar sentences and maximize the distance between sentence pairs that are semantically distant.

SetFit employs a two-step training approach: ST fine-tuning and classifier head training.

- In the *ST fine-tuning* step, an ST is fine-tuned on the input data in a contrastive, Siamese manner using sentence pairs. This contrastive fine-tuning approach enlarges the size of the training data in few-shot scenarios.

  Given a small set of K labeled instances $D = \{(x_i, y_i)\}$, where x_i and y_i denote sentences and their respective class labels, the method constructs R positive and negative triplets for each class label $c \in C$.
  Positive triplets $T_c^p = \{(x_i, x_j, 1)\}$ are created from pairs of sentences x_i and x_j randomly drawn from the same class c such that $y_i = y_j = c$. Conversely, negative triplets $T_c^n = \{(x_i, x_j, 0)\}$ are composed of sentences x_i from class c and sentences x_j randomly selected from different classes, ensuring $y_i = c, y_j \neq c$.
  The final contrastive fine-tuning dataset T is the result of concatenating positive and negative triplets from all class labels: $T = \{(T_0^p, T_0^n), (T_1^p, T_1^n), ..., (T_{|C|}^p, T_{|C|}^n)\}$, where $|C|$ is the number of class labels, and $|T| = 2R|C|$ gives the total number of pairs in T. Here, R is a hyperparameter which by default is set to 20 in all evaluations.

- The *classification head training* step involves training a text classification head using the encoded training data generated by the fine-tuned ST from the first step. In this step, a logistic regression model is used as the text classification head.

At inference time, the fine-tuned ST encodes an unseen input sentence and produces a sentence embedding. The classification head then produces the class prediction of the input sentence based on its sentence embedding.

SetFit has demonstrated strong results across multiple NLP tasks, even with a small number of training samples. It has outperformed both standard PLM fine-tuning and state-of-the-art methods such as ADAPET [61] and T-FEW [33] in a number of few-shot text classification tasks.

SetFit offers several advantages over comparable approaches. It's faster at both inference and training, requires much smaller base models, and alleviates the need for the instability and inconvenience of prompt crafting. Unlike many methods, SetFit is not subject to high variability from manually crafted prompts, and typically requires orders of magnitude fewer parameters than existing techniques to achieve high accuracy.

SetFit has shown robust performance as a few-shot text classifier in languages other than English and across varying typologies. It has also proven useful in few-shot distillation setups. In practical few-shot settings, SetFit provides a simple, prompt-free method for achieving high performance.

Strength and weaknesses

Iterative T-CALR is thought as a cold start method because it **exploits only a limited number of labeled examples** which would otherwise be insufficient to effectively train

the target transformer model. Thanks to SetFit, which finetunes SBERT in a contrastive learning manner, from a few examples per class it is possible to create a much larger training dataset.

In order to handle the constraint of limited labeled training data in few-shot scenarios more effectively, the ST fine-tuning phase leverages a contrastive training approach. This approach, commonly used for establishing image similarity, is adapted for text data.
This strategy effectively magnifies the size of the training data in few-shot scenarios. Given a small number of labeled examples (K) for a binary classification task, the potential size of the ST fine-tuning set T can be calculated from the number of unique sentence pairs that can be generated, specifically $K(K-1)/2$, which substantially exceeds the initial size of K. Therefore, in a limited resources scenario, instead of fine-tuning in an unstable way a standard transformer with k examples per class, to extract more meaningful embeddings a SBERT model is fine-tuned on $2R|C|$ examples.

However, this approach still suffers from the risk of selecting in a cycle similar examples between each other. Within a cluster the examples selected are still those with higher information density score, thus it could happen that queried examples are contain redundant information.

Ultimately, regarding efficiency, this method requires training a model as many times as the number of iterations. Consequently, examples can not be selected all at once, as is possible in the case of TCALR.

3.3.3 Balanced Cold Start (BCS)

An issue of active learning techniques is **biased sampling** towards majority classes. This problem is exacerbated in the case where the majority class is an "improper" class that collects whatever is not one of the "proper" classes. Data belonging to a proper class (the standard notion of class in machine learning) is usually characterized by a set recognizable of features and has different characteristics from other proper classes. On the other hand, an improper classes may not indicate a specific type instead it collects under its name many everything that is not classifiable as a one of the proper classes. It is often the case that majority of the dataset's outliers belong to an improper class. In the case of Altilia's dataset, *altro* is the improper class and the other eight classes (*visura ipotecaria*, *contratto di mutuo* ecc.) are the proper ones becasue they refer to a specific document type.

As identified by Chen et al. in [8] active learning techniques in a cold start scenario perform biased queries and ignore some classes, usually minority ones.
Here the assumption is that having a more **balanced initial pool** to kick-start the active learning process will yield more stable fine tuned model with better uncertainty measures and consequently more effective active learning cycles.

The proposed technique is employed in iterative cold start cycles where the first batch of examples is selected as with the TCALR technique in a one shot. Afterwards the sampling is performed by selecting the examples with the highest balance score per cluster. Basically the framework structure is the same as in TCALR. However, the score based on which examples shifts after the first iteration from a density score to a **balance score**.

Let D_L be a labeled pool consisting of n examples distributed across k classes $C = \{c_i|0 \leq$

$i < k\}$. Let's define a threshold parameter $\theta = \frac{n}{k}$, which represents the average number of examples per class. Then, a class c_i is defined as:

- A *majority class* if the number of examples in c_i is greater than or equal to θ, i.e., $|c_i| \geq \theta$.
- A *minority class* if the number of examples in c_i is less than θ, i.e., $|c_i| < \theta$.

This heuristic 3.4 is composed by two terms which concur at ranking examples based on expected contribution to class balance in the labeled pool. The first term of the expression assigns higher scores to those points which are likely to belong to underrepresented classes C_{min}, while the second term increases the score of the data points which are furthest from the majority classes C_{maj}.

$$\begin{aligned} \text{Balance score}(x_i) = & \sum_{c_k \in C_{min}} \left(\frac{\sum_{x_j \in c_k} sim(x_i, x_j)}{|c_k|} \times \frac{\theta}{|c_k|} \right) + \\ & \sum_{c_k \in C_{maj}} \left(\frac{\sum_{x_j \in c_k} diss(x_i, x_j)}{|c_k|} \times \frac{|c_k|}{\theta} \right) \end{aligned} \tag{3.4}$$

A work by Wertz et al. on balanced active learning for text classification [68] evaluated the efficacy of using as class representations the centroids of the labeled examples. They found out that the average of the embeddings is not an effective representation of the classes and they obtained a performance worse than random sampling.

Therefore, in this thesis work the likeness of an example to belong to a certain class is measured by using all the available labeled points, without averaging them or extracting from them a unique class representation.

The first term of 3.4 computes a weighted similarity score for the unlabeled example x_i for every minority class in C_{min}.
For each minority class c_k, it calculates the average similarity score of x_i with all examples in the class ($x_j \in c_k$). The similarity between x_i and each example x_j in the class c_k is given by the cosine similarity function.
This average similarity score is then multiplied by the term $\frac{\theta}{|c_k|}$. This term acts as a weighting factor that emphasizes the importance of underrepresented classes: the smaller the class size $|c_k|$ relative to the threshold θ, the larger the weighting factor.
As a result, minority classes that are substantially underrepresented (i.e., classes where $|c_k|$ is much less than θ) will have a greater influence on the balance score of x_i.
In essence, this term of the formula measures how well the unlabeled example x_i fits into each of the minority classes, with more weight given to classes that are significantly underrepresented.
The second term is a mirrored version of the first. In fact, for each class in C_{maj} it calculates the average dissimilarity which is just equal to $1 - sim()$ and the $\frac{|c_k|}{\theta}$ boosts the score based on how larger is the class with respect to the threshold.

Strengths and weaknesses

The BCS sampling technique aims at collecting an as balanced as possible initial pool. This approach supports the model's learning in the early stages by ensuring it captures features from underrepresented classes, which might otherwise be overlooked during the active learning cycles.

The BCS technique is broadly applicable, extending to a wide variety of datasets. It could also prove advantageous in situations where, despite a balanced prior class distribution, active learning selection remains biased towards certain classes. Another key strength of this heuristic is its independence from tunable hyperparameters, thereby avoiding potential drastic variations in behavior across different datasets.

However, the BCS technique does not come without its limitations. Specifically, data points that are close together in the embedding space tend to have similar balance scores. As a result, akin to the scenario with the information density score, this technique may select similar examples together. This could potentially lead to sub-optimal utilization of the budget, as it inadvertently reduces the diversity within the selected sample pool.

Chapter 4

Experiments

4.1 Cold start for pool initialization

This section sets out to test the effectiveness of various cold-start techniques in building a useful initial labeled set to kick start active learning cycles effectively.

4.1.1 Experiment Objectives and Research Questions

Unlike previous studies that primarily assess cold-start methods' performance against warm-start active learning approaches in low-resource settings, this study offers a different perspective. Cold start methods proposed in literature works, presented in 2.4.2, are tested against warm start active learning approaches based on their performance in the first few iterations and not by looking at how uncertainty based techniques could exploit the pools selected by cold start ones. Prior work typically involves active learning iterations starting with minimal or no labeled examples. Consequently, it tests the ability of a technique to create a labeled pool, which delivers the best performance. Because classic active learning methods rely on uncertainty measures provided by the target model, these are very unreliable and variable when training with little data, especially in the deep learning case. Previous results have indicated that while cold-start techniques tend to achieve higher accuracy faster, they are eventually caught up, and sometimes even surpassed, by warm-start methods in the later iterations.

Unlike previous analyses, the present experiment evaluates the 'goodness' of cold-start techniques based on the performance of the target model trained on the initial labeled set selected though cold start techniques and the constructed sets by warm start AL techniques in the subsequent iterations. In other words, a cold start technique works well if it is able to select an initial labeled pool such that the performance curve, along the iterations, of the model trained with active learning reaches the peak earlier.

An optimally sampled initial pool allows for a better-trained model that provides more reliable uncertainty measures, which are of substantial benefit to warm-start active learning methods. In this experiment various cold-start methods are employed to select an initial sample from which standard active learning iterations will commence.

This experiment seeks to answer the following research question:

- Do warm start active learning methods benefit from an initial sample selected through cold-start techniques? How long does the positive effect last?

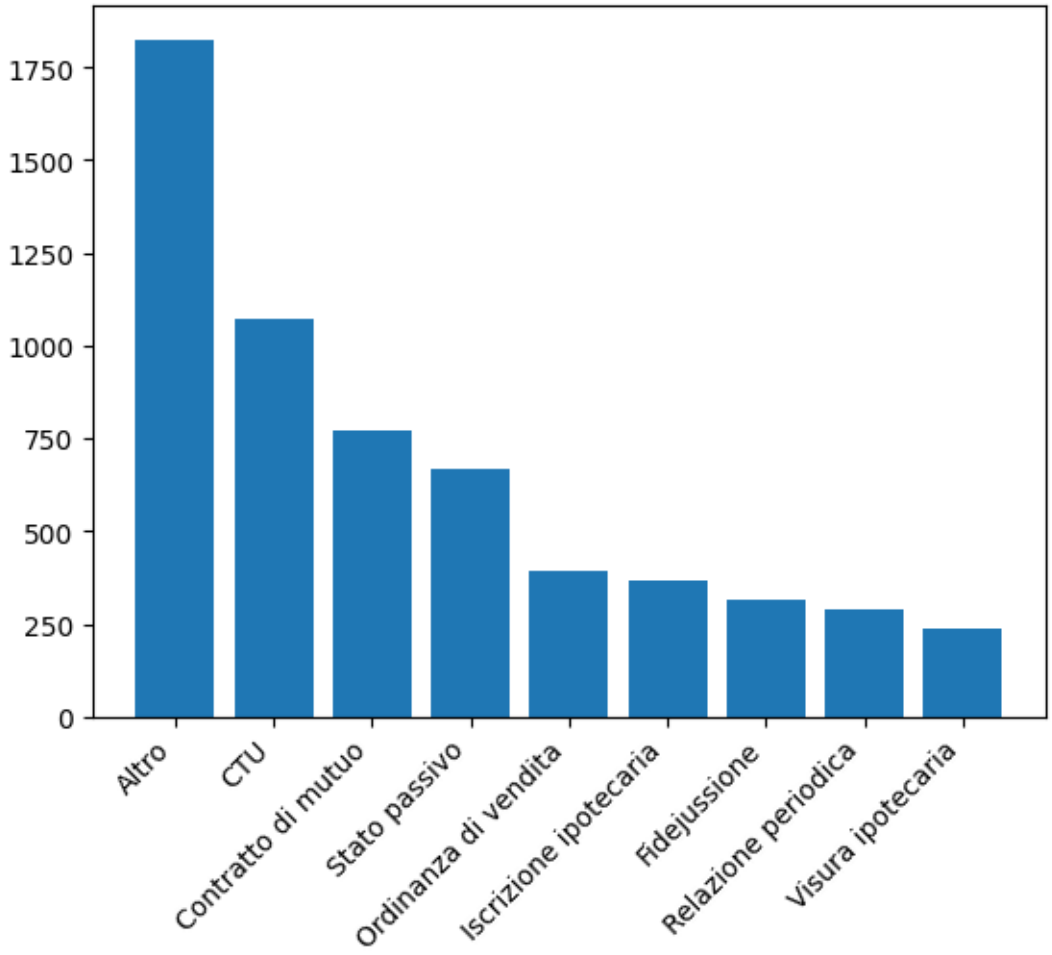


Figure 4.1: Altilia down-sampled version class distribution

4.1.2 Experimental Setup

Active learning experiments are more expensive, than standard training setups, in time and computational terms because training and sampling are performed repeatedly for a predefined number of iterations and training takes longer as the iterations increase. As a "trade-off" the original dataset 3.1 has been down-sampled to nearly 15% of its size. The class distribution of the 6000 examples is shown in figure 4.1. Moreover, the model techniques' performance at each active learning cycle is evaluated on a 3-fold split of the data. As in the preliminary works experiments, the fold splits are done based on the documents. Therefore, the class distribution across splits is not exactly the same.
The train set is used in the active learning simulation by masking at the beginning all the labels, thus considering it as an unlabeled pool.

At each active learning cycle the target model is trained on the enlarged labeled set and is evaluated on the development set of the corresponding fold and then the averaged results across folds will be shown. Evaluation measures include: weighted-f1, macro-f1, micro-f1, accuracy, f1, precision and recall. The metrics' details are explained in section 3.2.1. Through these metrics both overall and per class performance is captured to better understand strengths and weaknesses of applied techniques.

Experiments are executed on the google cloud platform Colab[1], results are tracked through the Wandb platform[2] and used pre trained models are available on Hugging Face[3].

4.1.3 Baseline Techniques

To better understand and evaluate the effectiveness of proposed cold start techniques some baselines are employed.

First, the target model is trained on **all available data** to see metrics across iterations in the perspective of the maximum score possible.

Second, employed in every active learning work published, **random sampling** is used as a lower bound. If a technique performs worse than random sampling this gives no reasons to use a more complicated technique.

Third, **random** sampling is applied **within the clusters** created by BIRCH. This baseline is used to test the effectiveness of the representative by information density.

Finally, to test the hypothesis that balance is an important characteristic of an effective initial labeled pool, a **simulated balanced** initial sample is tested along other techniques. By simulated it is meant that the selection of examples in a balanced way is done by looking at the ground truth labels, which are not normally available in active learning cycles.

4.1.4 Implementation Details

The features and the objectives of the proposed methods are described in the methodology chapter 3.3. In this section, for each technique included in this comparison experiment, the practical implementation details are illustrated.
The experiment is divided in two parts: cold start active learning and warm start active learning. In the first part, each CS method samples 396 examples, while in the second part, standard AL cycles are performed with BADGE starting from initial pool selected during the first phase. Therefor, the variable is the cold start method and its initial selection of 396 data points.

Why exactly 396?
In the preliminary experiments on active learning (see 3.2.2 the comparison between different AL techniques was performed by using a randomly sampled initial pool of 400 examples. Ideally the sample size would be the same also in this case but for how the techniques are designed this is not possible. Each technique involves the even sampling from each of the clusters created by BIRCH, the number of clusters is decided apriori equal to the number of classes present in the dataset (9), thus the nearest integer to 400 that is also a multiple of 9 is 396.

In general, the distiluse-base-multilingual-cased-v1[4] is used as the pre-trained SBERT model for sentence embeddings in the cold start phase. This is a multilingual model

[1] https://colab.google/
[2] https://wandb.ai/site
[3] https://huggingface.co/models
[4] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1

able to handle 15 different languages, including Italian, allows a maximum input size of 128 tokens and maps sentences to a 512 dimensional dense vector. It is based on the DistilBertModel[50] architecture with a mean pooling layer on top.
For the clustering part, the open source implementation of BIRCH available on SickitLearn[5] is used with its default hyperparameters: threshold equal to 0.5 and branching factor to 50. To assign the information density scores the code made available on modAL[6] is utilized. For what concerns the AL phase code implementation is the same as in the preliminary work 3.2: ilberto-uncased-from-camembert[7] is the trained target model and an adapted version of the code published in [3] is used for BADGE sampling.

The initial pool size sampled is the same for every method but the intermediate steps differ. In the case of T-CALR sampling can be performed in one shot: from each cluster the 99 data points with the highest information score are selected. Iterative T-CALR and BCS operate in 4 iterations in which. In the first iteration the samples are selected with the T-CALR technique because there is no information about the labels yet: the SBERT model cannot be fine tuned through SetFit and balance scores cannot be calculated yet. The remaining 297 examples are selected in the next 3 iterations through the specific technique.

In iterative T-CALR the SBERT model is finetuned, thus the embeddings, clusters and information density scores change and the top examples from each cluster are selected. SetFit fine tuning is happens in two steps: first the body is trained for 1 epoch with cosine similarity loss then both body and classification head are trained together for 25 epochs with cross entropy loss. The parameter R which determines the number of positive/negative pairs to train the body is set to 20. Body learning rate is set to 1×10^{-5} while for the head a value of 1×10^{-2} is set. With BCS the 11 examples from each cluster with the highest balance score (see 3.3.3), given the acquired class distribution, are queried.
T-CALR with random selection is implemented as T-CALR without the information density ranking part, that is substituted by random sampling from each cluster. Finally, the simulated balance is implemented through even random sampling within each class pool.

In the second phase, 15 AL cycles, at each iteration 100 samples are selected with the BADGE sampling technique, queried and added to the labeled pool. Moreover, the target model is trained from scratch at each iteration on the updated labeled set. The training hyperparameters are the same used in the preliminary work 3.2: 5 epochs using a learning rate of 5×10^{-6}, a batch size of 16 samples and optimization through AdamW [35] optimizer.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html
[6] https://github.com/modAL-python/modAL
[7] https://huggingface.co/idb-ita/gilberto-uncased-from-camembert

4.1.5 Results

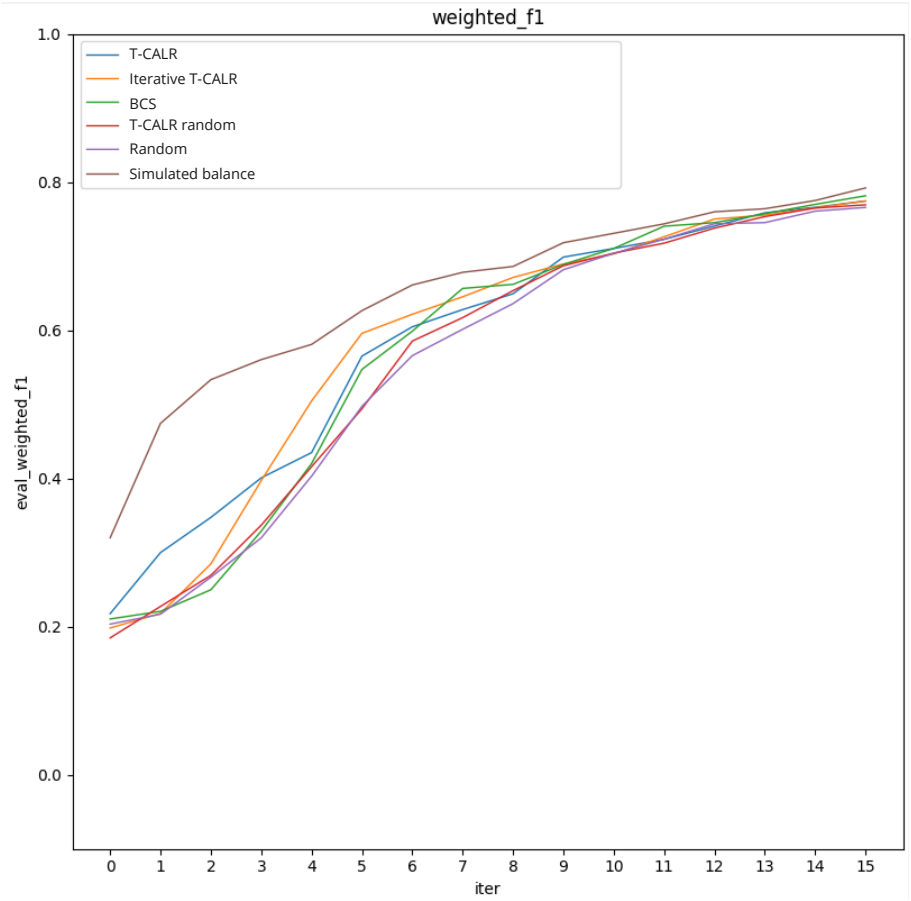


Figure 4.2: Averaged on 3-folds weighted f1 score for each active learning cycle

In figure 4.2 are reported the weighted f1 scores of the active learning cycles starting from the different initial samples.
Each cold-start technique provides a unique initial pool of samples to begin the active learning cycles. At the initial iteration (iteration 0), proposed techniques result in a weighted f1 score around 0.2, indicating similar effectiveness in their respective sampling strategies. Of these, T-CALR demonstrates a superior weighted f1 score of 0.22, while T-CALR with random sampling falls short with a score of 0.18. Simulated balance achieves a significantly higher score of 0.32, making it the best performing cold-start technique in this experimental setting.

As the active learning process continues across subsequent iterations, simulated balance maintains its top-performing status. While the performance gap narrows as iterations increase, simulated balance consistently outperforms other techniques throughout the active learning process. Among the other techniques, T-CALR delivers a superior performance from iterations till iteration 3 where is surpassed by Iterative T-CALR (weighted f1 of 0.4) that takes the lead until approximately the 8th iteration (weighted f1 of 0.67). Balanced cold start and random sampling initially show the weakest performance, but they manage to catch up by the 9th iteration.

The upper-bound performance, as determined by training on all available data, is a weighted

f1 score of 0.83. In comparison, by the 15th iteration, simulated balanced cold-start approaches this upper bound, reaching a weighted f1 score of 0.8. This achievement, derived from only nearly half of the training data, underscores the potential efficiency of an effectively sampled initial pool in the active learning process.

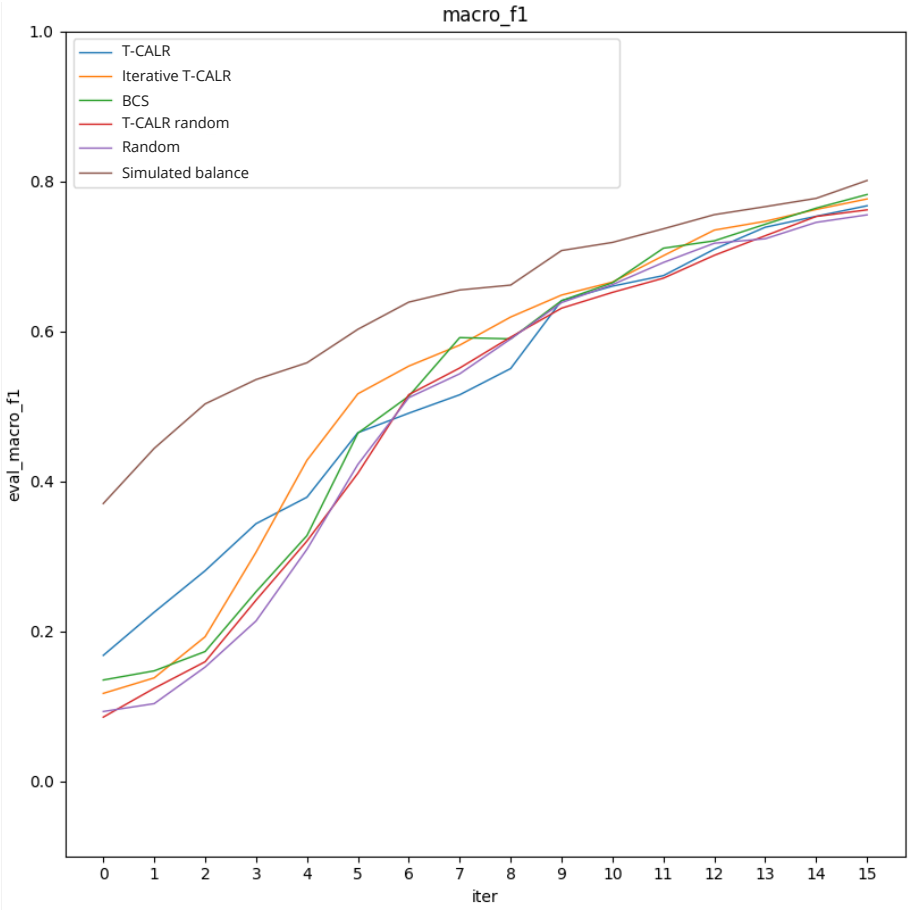
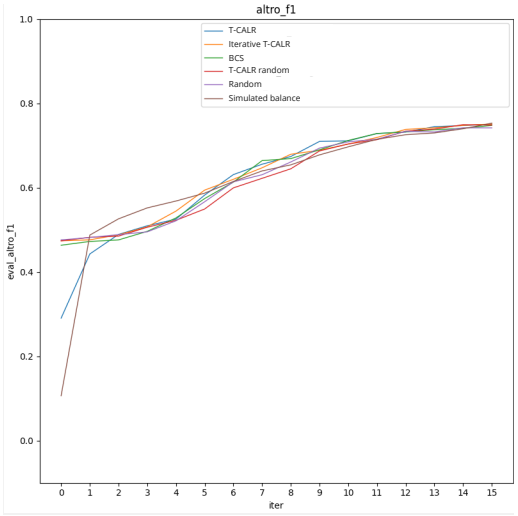
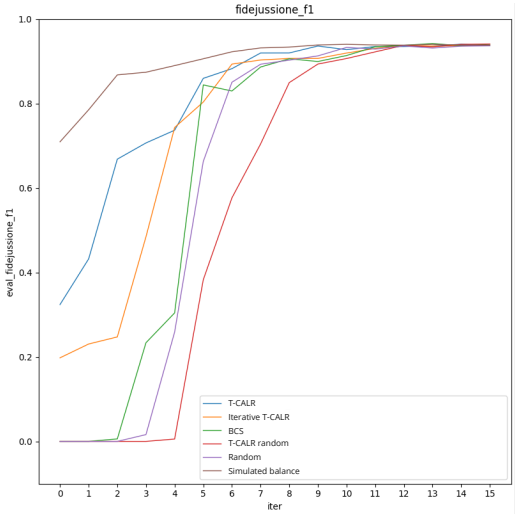


Figure 4.3: Averaged on 3-folds macro f1 score for each active learning cycle

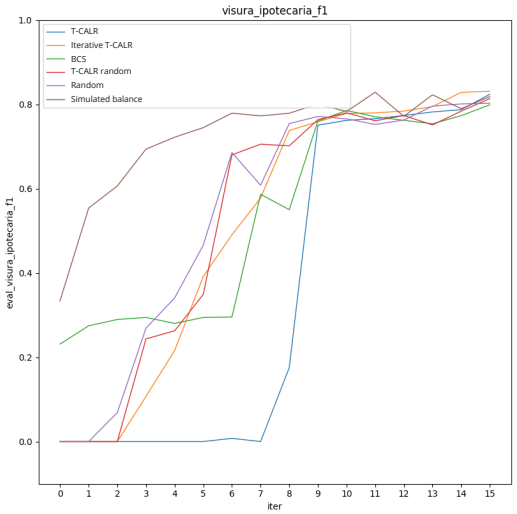
In figure 4.3 is reported the performance comparison taking into consideration the macro-f1 score. The trend and methods ranking along the iterations remains the same as with the weighted-f1. What can be noticed here is a larger gap between the lines, i.e. greater performance differences because this metric does not account for the class frequencies in the evaluation set when averaging the f1 scores per class.



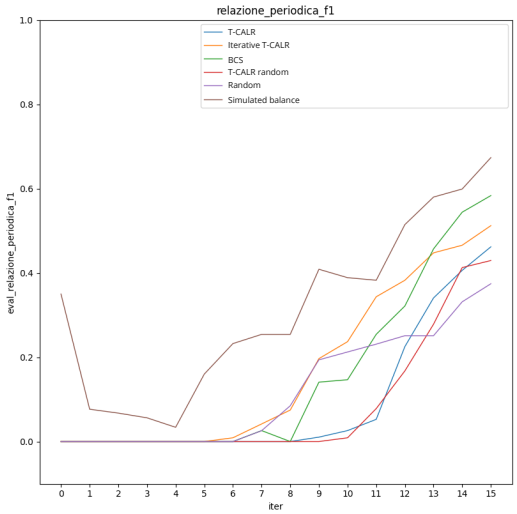
(a) Averaged on 3-folds f1 score of class *altro* for each active learning cycle



(b) Averaged on 3-folds f1 score of class *fidejussione* for each active learning cycle



(c) Averaged on 3-folds f1 score of class *visura ipotecaria* for each active learning cycle



(d) Averaged on 3-folds f1 score of class *relazione periodica* for each active learning cycle

Figure 4.4: Class f1 metrics across active learning cycles. Remaining class metrics can be found in appendix (see A.2)

To gain a more nuanced understanding of the different method performances within a multi-class classification task, it is beneficial to assess class-specific metrics. Figure 4.4 presents the results for four specific classes: *altro*, *fidejussione*, *visura ipotecaria*, and *relazione periodica*.

The *altro* class demonstrates a convergence of performance across all methods following the initial iteration, with the simulated balance and T-CALR methods showing slightly weaker results at iteration 0.
For the remaining three classes, a consistent trend is the superior performance of the simulated balance method. In Figure (b), the T-CALR with random selection method notably

underperforms. In the *visura ipotecaria* graph, the F1 score for T-CALR remains at 0 until the 8th iteration, indicating a challenge for the model to learn this class with this method. Most strikingly, the *relazione periodica* class proves the most challenging for the models to learn, except when the simulated balance method is used. With other methods, the models seem to begin learning only at the 7th iteration when 1100 examples are present in the labeled pool.

4.1.6 Analysis and Discussion

The macro f1 score aggregates the f1 scores of the different classes by averaging on the number of classes, without considering their sizes in the development set as the weighted f1 score does. Therefor, it reflects better the actual performance of the model on the different labels in this imbalanced dataset case. The figure 4.3 reporting the macro f1 scores highlights the fact that a balanced initial pool yields better results also for the less represented classes. In fact, the macro f1 score is higher for Simulated Balance and lower for the other techniques w.r.t. weighted f1.

Only the Simulated Balance initial pool achieves an f1 score greater than zero for all classes from the first iteration. This appears to be a crucial characteristic that helps the following active learning iterations. Having a balanced "view" of all the classes allows BADGE to express better its uncertainty when doing inference on unlabeled samples, consequently selecting more effective examples.

Unexpectedly, despite the fact that some methods construct an initial labeled pool that is more balanced than random sampling, it appears to be not enough to provide a performance near the Simulated Balance sampling. To measure class balance (**B score**) of a set of data we use the concept of entropy:

$$\text{B score} = \frac{H}{\log k} = \frac{-\sum_{i=1}^{k} \frac{c_i}{n} \log \frac{c_i}{n}}{\log k} \tag{4.1}$$

where n is the size of the set, k the number of classes and c_i the number of examples belonging to that class present in the set.

In table 4.1 are reported the class distribution B scores of the first fold initial pools while in table 4.2 there are the averaged scores over 3 folds. It can be seen that both BCS and Iterative T-CALR result in initial pools with a B score, higher than random, of 0.94 and 0.92, on average respectively. A problem that emerges is the excessive sampling from the *altro* class which does not appear to be a problem with T-CALR sampling. However, T-CALR struggles to select data points from the minority classes (*relazione periodica* and *visura ipotecaria*) which determine a very low f1 score for many active learning iterations (see 4.4. From these tables it must be noted that BCS actually achieves the best balance score within the proposed techniques. Moreover, Iterative T-CALR, even if it does not address the balance issue directly, it is able to obtain label diversity of the sampled points. This can be attributed to its ability to refine the embeddings and define better clusters.

4.1.7 Conclusions

In this experiment has been tested the effect of the different initial pools selected by cold start techniques. Results show that class balance is crucial, not only allows to kick start

Strategy	Altro	CdM	CTU	Fid	II	OdV	RP	SP	VI	B score
T-CALR	63	64	85	32	59	45	5	42	1	0.89
Iterative T-CALR	101	48	59	31	51	33	6	43	24	0.93
BCS	115	41	54	28	32	41	28	45	12	0.92
T-CALR random	116	66	76	11	21	43	17	40	6	0.87
Random	126	63	68	27	25	25	10	44	8	0.87
Simulated B	44	44	44	44	44	44	44	44	44	1

Table 4.1: Class balance scores (B score) of the initial sampled pool by cold start techniques in fold 0. CdM: *contratto di mutuo*, Fid: *fidejussione*, II:*iscrizione ipotecaria*, OdV: *ordinanza di vendita*, RP: *relazione periodica*, SP: *stato passivo*, VI: *visura ipotecaria*

Strategy	B score
T-CALR	0.89
Iterative T-CALR	0.92
BCS	0.95
T-CALR random	0.88
Random	0.88
Simulated B	1

Table 4.2: Averaged over 3 folds balance scores (B score) of different strategies

the active learning cycles with higher performance but also to reach higher evaluation score with less examples. However, balance is not the only factor that affects the initial pool effectiveness. In fact, other techniques aiming to create a more balanced dataset do not yield equally positive results.

T-CALR and Iterative T-CALR produce less balanced pools than BCS but the results obtained starting from those sets are better in the initial iterations. They obtain significantly better results in the first iterations thanks to the typical data selection process carried out through representative sampling.

The effect of the CS method disappears after selecting 1000 examples. Only Simulated Balance maintains a margin also at the end. This shows how important is a balanced training set in the context of very unbalanced data pools.

What can be learned from it?
In the context of active learning with an imbalanced unlabeled pool typical data selection and class balance are key features that the initial pool must have at the same time. The proposed techniques approached one key factor at a time. To be more precise, actually BCS employs a first round of representative data selection but does not implement a func-

tion that at the same time rates the "amount of balance" and "typicality" that a specific data point carries, consequently it does not perform as expected.

For future work in this direction two ideas could be explored:

- combine BCS with T-CALR. As simple as it could be, rank data points within a cluster with a score given by the product of information density and B score
- an iterative version of the first proposal. The embedding space can be updated by training the SBERT model in a contrastive manner yielding better clusters.

Limitations of this study include a 3-fold cross validation which is not ideal, some state of the art cold start techniques have not been tested on this data and the size of the initial pool was set apriori which may make a difference in the active learning cycles. This last point is the subject of the next experiment.

To summarize and answer to the related research question, this experimental evaluation reveals that an effective initial pool can significantly boost the early performance of warm-start active learning. However, the benefits tend to reduce over iterations, with the performance differences among various cold-start techniques shrinking over time. Consequently, while an ideal initial pool can give active learning a good start, achieving this ideal initial pool remains a challenging task, and the benefit it offers may become less significant with more iterations.

4.2 Cold vs Warm Start Active Learning Threshold

4.2.1 Experiment Objectives and Research Questions

The primary objective of this experiment is to investigate an understudied aspect of cold start (CS) active learning (AL). Previous experiments have examined whether cold start techniques could construct initial labeled pools that enhance the model's performance in subsequent warm start active learning cycles. However, these experiments set the size of the initial pool apriori, limiting their understanding of the full potential of CS techniques in the presence of larger pools. The aim here is to determine the optimal point to transition from cold start to warm start techniques during the active learning process.

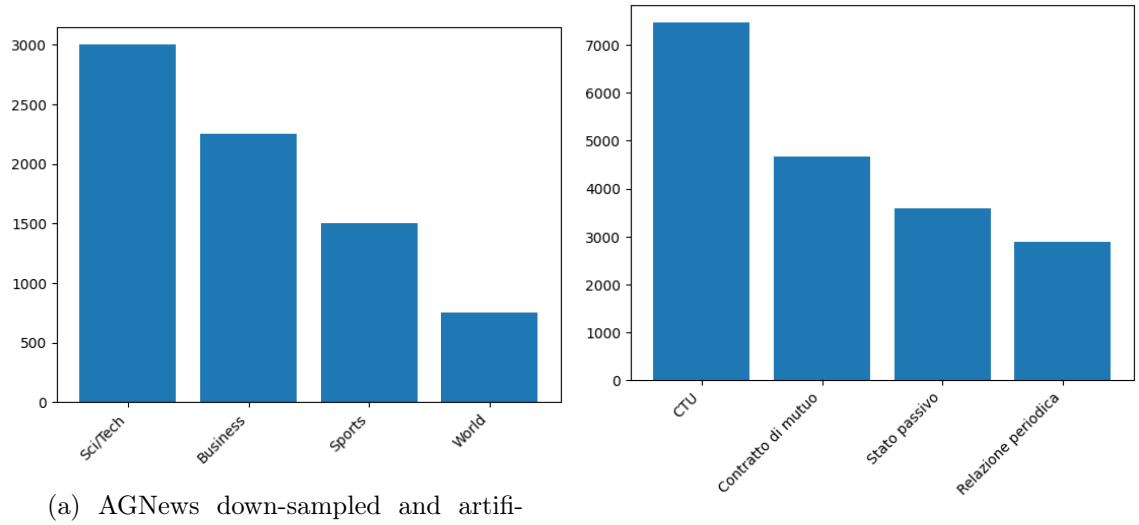
This experiment is designed to answer the following research question:

- At what point during the active learning process does warm start active learning surpass cold start active learning in terms of effectively sampling data points?

Based on prior results, it is hypothesized that cold start methods will outperform warm start methods in the early stages of active learning due to their ability to sample more effective data points without relying on uncertainty measures. However, it is anticipated that the advantage of cold start methods will diminish over time and at a certain point, warm start methods will become more effective.

The following metrics will be used to evaluate the performance of the active learning methods and answer our research question: weighted f1, macro f1, micro f1, accuracy, and for each class precision, recall, f1.

The rationale for this research question lies in the literature gap and the limitation of



(a) AGNews down-sampled and artificially imbalanced dataset

(b) Altilia down-sampled dataset

Figure 4.5: Datasets' class distribution

the previous experiment (4.1). Existing studies on cold start tend to compare CS methods with warm start ones in standard active learning cycles, which is not entirely fair or practical since CS active learning, by definition, does not utilize the target model's uncertainty—a critical element in efficient data selection during the later stages of active learning.

If this experiment successfully identifies a pattern that can indicate when to transition from cold start to warm start active learning techniques, it could have significant implications for the broader field of active learning, making the active learning process more efficient and robust.

4.2.2 Experimental Setup

This experiment is carried out using two different datasets: the first one provided by Altilia and the second one is the AGNews dataset. The AGNews dataset, widely used in text classification and active learning literature, contains 120k train examples, 7600 test examples and four classes. This training set has been down-sampled to a total of 7500 examples, inducing class imbalance to mimic the imbalance found in Altilia's dataset.

To further provide a comparison between academic and real-world data, also Altilia's dataset, which contains 9 classes, has been manipulated to only include the 4 most common classes, excluding the 'altro' class. This results in a similar dataset structure to AGNews but preserves the real-world noise and outliers. The datasets class distribution is reported in figure 4.5 below. The experimental conditions involve comparing cold start techniques proposed in the methodology section with BADGE active learning. Random sampling is used as a baseline, and an upper bound is established by training the target models on all available data. The granularity of our analysis is determined by the iterations, which are based on the size of the queried batches. We consider the "point" at which warm start active learning becomes more effective than cold start active learning as when the performance of the target model trained on the data selected by warm start is higher.

Consistency and reliability are ensured by fixing the random seed for the clustering part,

random sampling, and dataset shuffling.

The number of iterations is mainly determined by time constraints and based on the previous experiments' results, where it was observed that peak performance could be reached within these iterations.

In conducting the statistical analysis of the experiments, a stringent approach was adopted. Error bars, which were calculated using the standard error deviation (std) from a 5-fold cross-validation, were incorporated into the figures to depict the variability in the results. Additionally, various statistical tests were carried out to ascertain the significance of the observations.

Mood's median test [45] was employed, serving as a non-parametric alternative that helps verify the equality of medians across multiple groups. Additionally, the Mann-Whitney U rank test [37] was utilized. This commonly applied non-parametric method allowed for the comparison of two independent samples, evaluating the null hypothesis that the samples are derived from the same population.

A p-value of 0.05 was selected as the threshold for statistical significance, indicating a 5% risk of falsely rejecting the null hypothesis. Through the use of these rigorous statistical measures, it was ensured that the results obtained are robust, reliable, and valid. This comprehensive analysis lays the groundwork for further discussion and interpretation of the findings.

All experiments are implemented using Google Colab, PyTorch, and the Hugging Face libraries.

4.2.3 Implementation Details

In the paragraphs below is explained in detail how the experiment is organized, which sampling techniques are used and how they are compared.
In this study, we aim to evaluate the performance of different active learning processes, including baseline methods, all starting from the same initial conditions. Specifically, every process begins with an identical labeled pool of 100 examples selected using the T-CALR method. The main purpose of this design is to ascertain the behavioral patterns of various techniques when they originate from a common starting point, even in the case of random sampling. The choice of T-CALR as the starting method is dictated by its relevance as the foundational approach for the techniques proposed in the methodology section of this research.

After the selection of the initial pool of 100 examples, active learning iterations are performed using various cold start techniques. During each iteration, a query is made for 100 examples. Importantly, two distinct labeled sets are constructed at each iteration.
The first set consists of all examples selected up until the current iteration via the cold start method. In contrast, the second set is composed of examples chosen by the cold start method up until the previous iteration, along with a batch of examples selected with the warm start technique during the current iteration. This methodology allows for a comparative evaluation of the different techniques within the same active learning cycle.

Algorithm 4 Cold start vs Warm start

$N \leftarrow$ *total number of iterations*
$f() \leftarrow$ *target model*
$t = 0 \leftarrow$ *current iteration*
Let $D_L^{cs}(t) = \text{T-CALR}\,(D_U(t), 100)$, $|D_L^{cs}(t)| = 100$
Let $D_L^{ws}(t) = \text{T-CALR}\,(D_U(t), 100)$, $|D_L^{ws}(t)| = 100$
$D_U(t+1) = D_U(t)/D_L^{cs}(t)$
$f(D_L^{cs}(t), f(D_L^{ws}(t) \leftarrow$ *train and evaluate*
$S_{cs}() \leftarrow$ *cold start method*, $S_{ws}() \leftarrow$ *warm start method*
repeat
    $t+ = 1$
    $Q^{cs} = S_{cs}(D_U(t), 100) \leftarrow$ *queried examples by CS*
    $Q^{ws} = S_{ws}(D_U(t), 100) \leftarrow$ *queried examples by WS*
    $D_L^{cs}(t) = D_L^{cs}(t-1) + Q^{cs}$
    $D_L^{ws}(t) = D_L^{cs}(t-1) + Q^{ws}$
    $f(D_L^{cs}), f(D_L^{ws}) \leftarrow$ *train and evaluate*
    $D_U(t+1) = D_U(t)/Q^{cs}$
until $t < N$

A total of 19 active learning iterations are performed with each cold start technique. In each cycle, 100 examples are queried. However, at each iteration two labeled sets are constructed. The first one is given by all the selected examples till that iteration with the cold start method, the second one is given by the examples selected by the cold start method till the iteration before plus the batch of examples selected with the warm start technique at the current iteration.

The process is illustrated in pseudo-code (see 4). In this way, at each iteration the two labeled sets differ only by 100 examples. Along the active learning cycles only the examples queried by the examined cold start technique S_{cs} are accumulated. Therefor, at iteration t, the two labeled sets (D_L^{cs}, D_L^{ws}) are the same except 100 examples. In one case they are selected by S_{cs} in the other case by S_{ws}. In this way, it can be captured the effect of the newly acquired samples and understand when a method becomes better than the other. In other words when does the warm start method have enough examples to train such a model that can output useful uncertainty measures to sample the least confident ones. For this experiment as S_{ws} the BADGE technique is employed. Algorithm 4 is applied with different cold start techniques, i.e. different S_{cs}. The methods used are T-CALR, iterative T-CALR, BCS and as baselines simulated balance and random sampling.

Another sort of baseline is obtained by running a complete AL process using BADGE, where the labeled examples accumulated during the iterations are those selected by the warm start technique. It is just "standard" AL using BADGE that starts from the same 100 labeled examples selected with T-CALR used in the aforementioned process. This baseline is useful to understand how does BADGE behave if applied from the very beginning, when very few labeled examples are available.

Finally, for clarity, cold start techniques after embedding the examples with the corresponding SBERT model, 4 clusters (given that there are 4 classes) are constructed and from each cluster 25 examples are queried. For the AGNews dataset, as a pre-trained SBERT model *all-mpnet-base-v2*[8] is used while the target model trained is *bert-base-uncased*[9]. The models used for Altilia's data are the same as in experiment 4.1.

4.2.4 Results

In this section, the results of the active learning experiment conducted on two distinct datasets are examined and discussed. The first dataset, provided by Altilia, consists of business documents and specifically focuses on the four majority classes. This dataset exhibits a class imbalance, which presents unique challenges for the active learning processes. The second dataset, AGNews, is an academic text classification dataset comprising four classes, which have also been manipulated to showcase imbalance.

The performance of each method, which includes both cold start techniques (T-CALR, Iterative T-CALR, BCS) and baseline methods (Simulated balance, Random sampling, BADGE), is assessed over a series of 19 iterations (with the exception of Iterative T-CALR which is assessed over 9 iterations). The primary metrics used to evaluate these methods are the weighted and macro F1 scores, providing a balanced measure of each method's precision and recall. Additionally, the performance of each method per class is also plotted to visualize class-specific trends and behaviors.
An overarching observation that becomes apparent during the experiments is that the performance of BADGE sampling method consistently surpasses other methods, becoming the highest by iteration 4 for both datasets. Conversely, T-CALR method exhibits the poorest performance across the boards.

For each dataset we report the experiment results with two types of bar plots. The first one (figures 4.6 and 4.8) provides the comparison between cold start methods, baselines and BADGE. In the second type (figures 4.7 and 4.9) are reported also the performance at each iteration given by the batch sampled through the warm start technique, i.e. comparison between the performance given D_L^{cs} and D_L^{ws}.

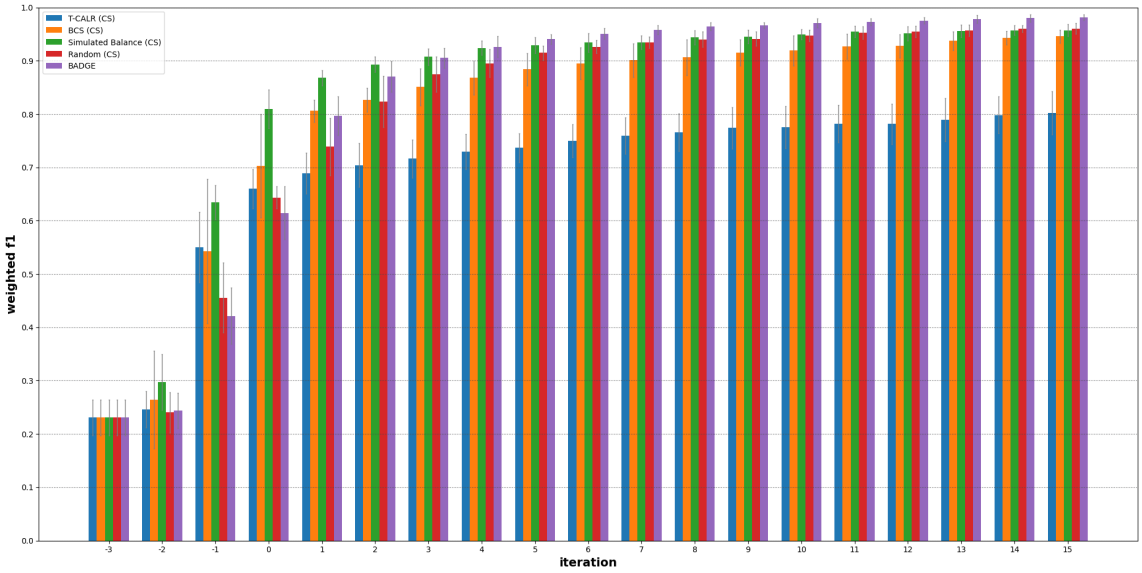
Altilia dataset results

The results from the active learning experiment conducted on the Altilia dataset offer intriguing insights into the performance of various cold start techniques and baseline methods.

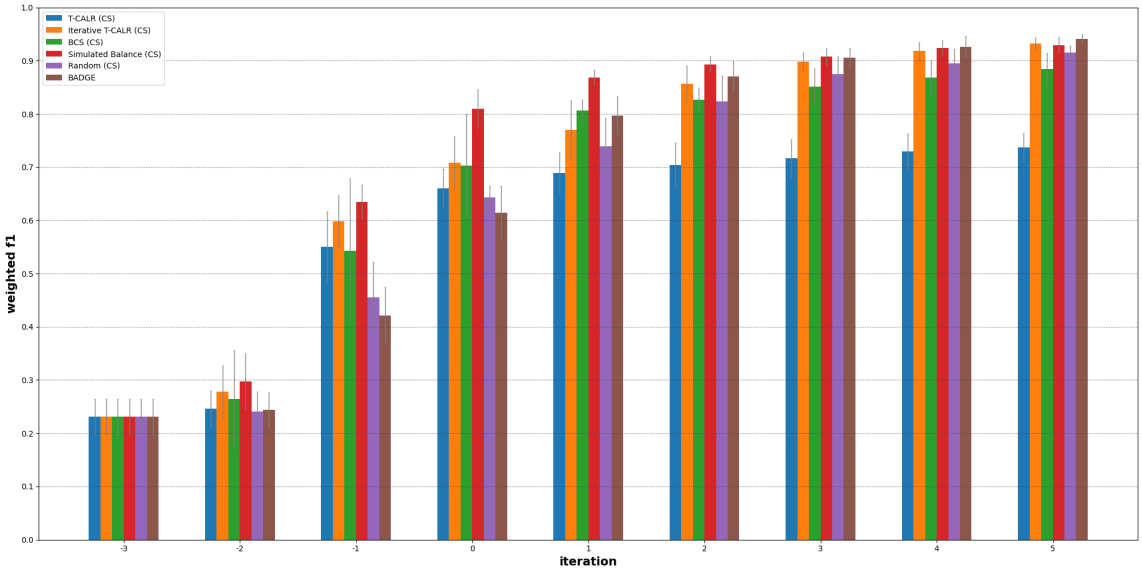
From the fourth iteration onwards, the warm start technique BADGE emerges as the most proficient sampling method, achieving a weighted F1 score of 0.95. This score surpasses even that of the simulated balance, which scores 0.935. Furthermore, the Iterative T-CALR method proves to be the most effective among the proposed cold start methods after the second iteration. In contrast, T-CALR consistently underperforms, showing only minor score improvements on the evaluation dataset after the initial iteration.

When comparing the different cold start techniques, each one demonstrates distinct performance patterns. T-CALR notably struggles with the class "relazione periodica", achieving

[8] https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[9] https://huggingface.co/bert-base-uncased



(a) 19 AL iterations

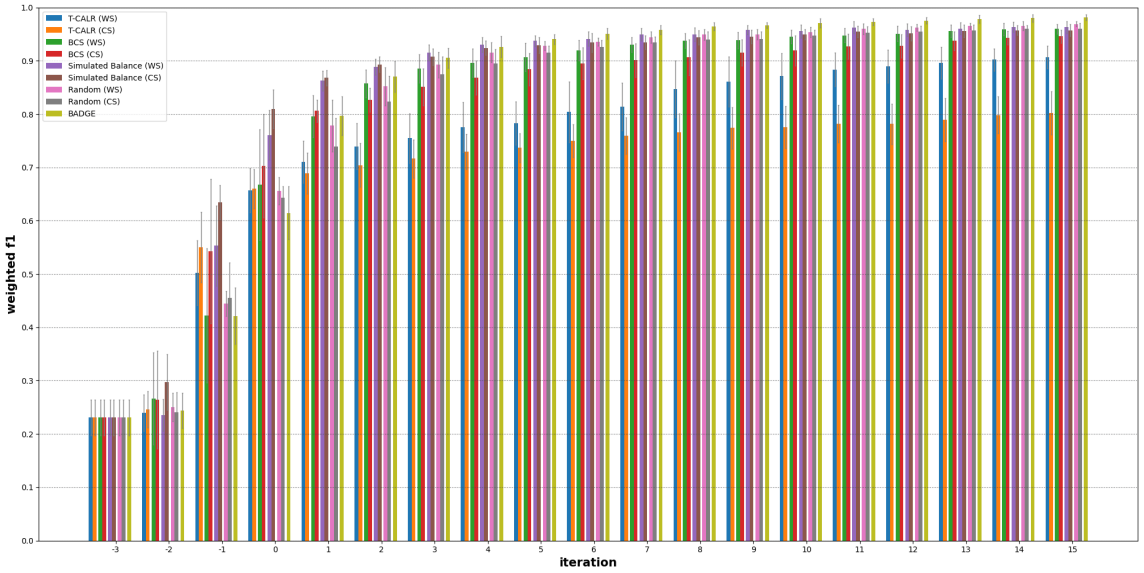


(b) 9 iterations with all methods

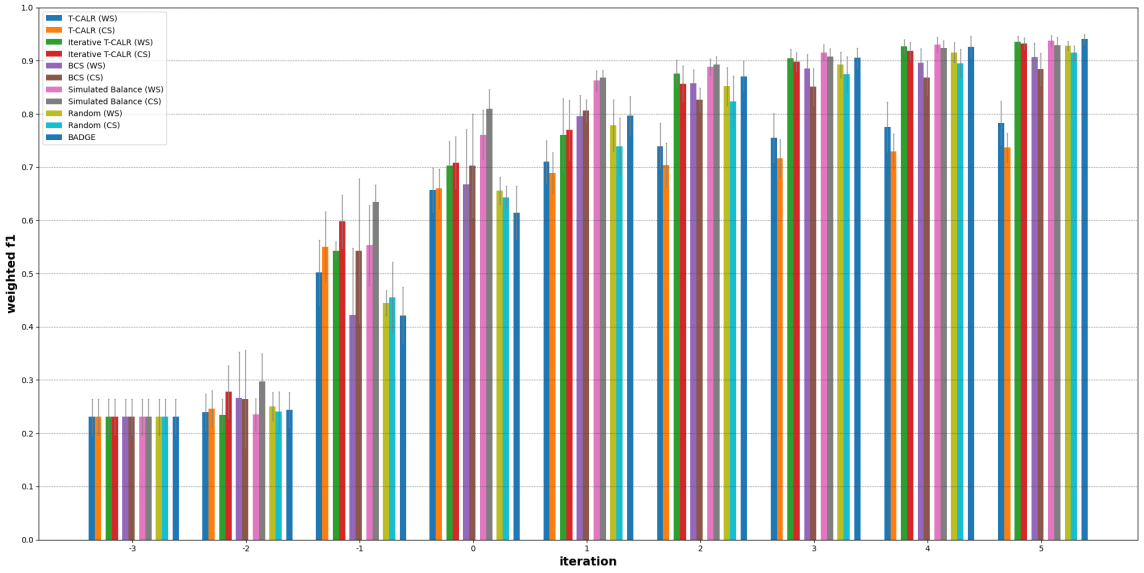
Figure 4.6: The average and std on 5-folds of the weighted f1 score on Altilia dataset. The last bar on the right is the BADGE baseline. Note figure a) on the right does not include Iterative T-CALR

only a 0.28 performance score at the fifteenth iteration. Additionally, the batches selected by T-CALR for active learning cycles indicate that the BADGE method provides a greater performance boost from the first iteration onward.

Iterative T-CALR excels in comparison, reaching a performance comparable to simulated balance and BADGE by the third iteration. Despite initial difficulties with the class "relazione periodica", Iterative T-CALR surpasses BCS at the third iteration and simulated balance at the fifth iteration. Interestingly, the performance boost from the batches selected by the BADGE technique outpaces the Iterative T-CALR from the second iteration, albeit by a small margin.



(a) 19 AL iterations



(b) 9 iterations with all methods

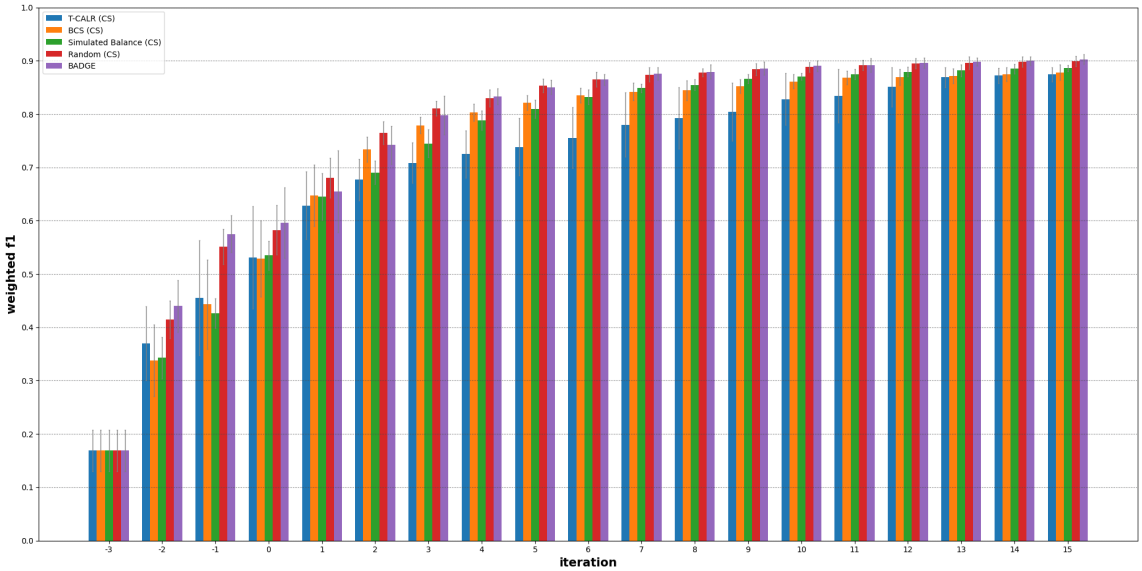
Figure 4.7: The average and std on 5-folds of the weighted f1 score on Altilia dataset. For each tested method (marked with CS) the bar on its left (marked with WS) represents the performance given D_L^{ws}. The bar line on the right is the BADGE baseline. Note figure a) on the right does not include Iterative T-CALR.

BCS surpasses T-CALR in performance but falls short compared to Iterative T-CALR and random sampling, which outperform it from the second and third iterations, respectively. BCS holds the best performance among the proposed CS techniques until the second iteration for classes "relazione periodica" and "stato passivo". However, the effectiveness of the batches sampled by BCS is overtaken by the BADGE method at the second iteration.

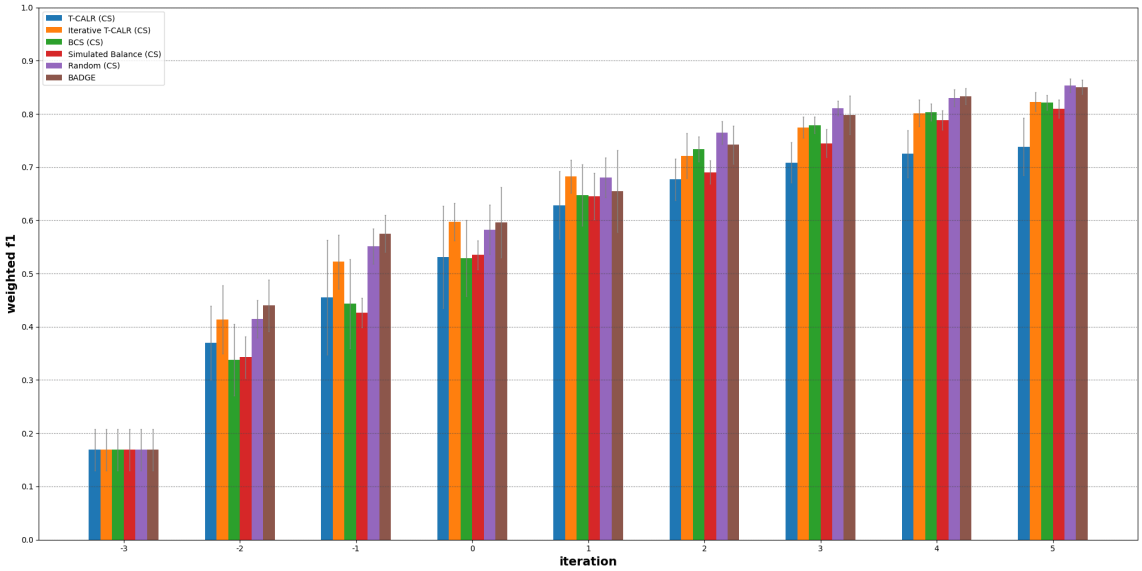
Consistently, simulated balance outperforms other cold start methods and baselines until the third iteration, after which it is surpassed by BADGE. As of the fifth iteration,

BADGE consistently achieves the highest performance by a considerable margin. Furthermore, the random sampling method surpasses both BCS and T-CALR from the third iteration onward.

AGNews dataset results



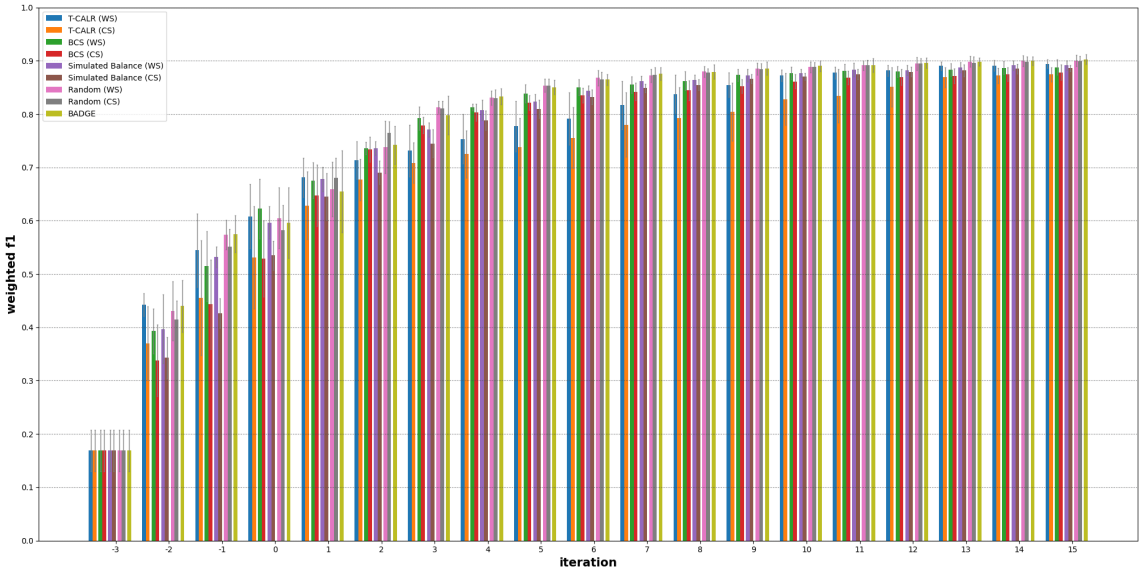
(a) 19 AL iterations



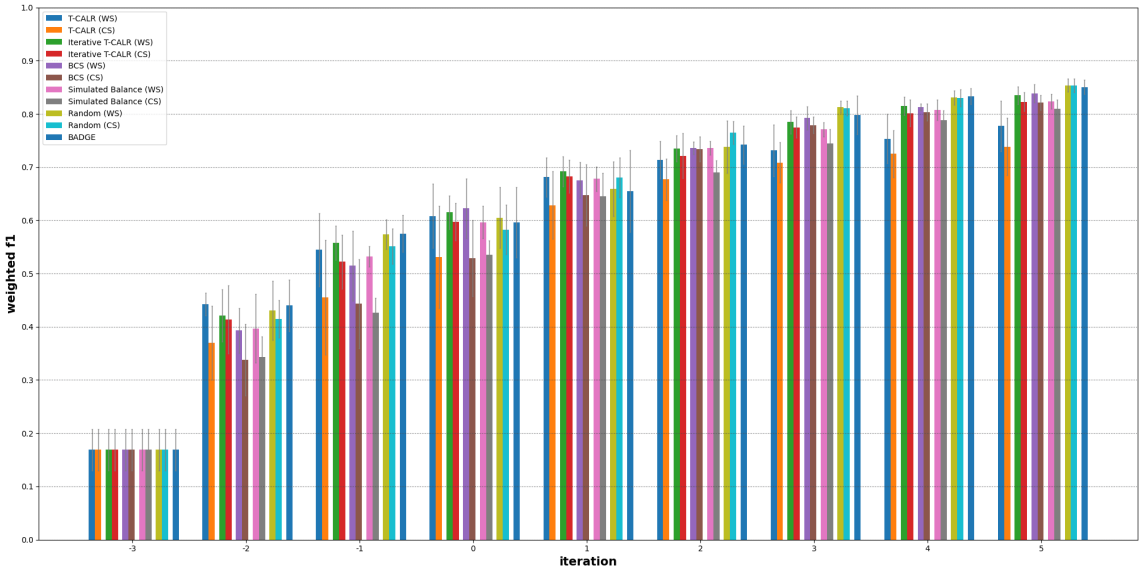
(b) 9 iterations with all methods

Figure 4.8: The average and std on 5-folds of the weighted f1 score on AGNews dataset. The last bar on the right is the BADGE baseline. Note figure a) on the right does not include Iterative T-CALR

One key observation from the experiments is the unexpectedly high performance of random sampling in the weighted-F1 bar plots. While BADGE sampling performance is either on par with or outperforms random sampling, T-CALR consistently underperforms after the initial few iterations.



(a) 19 AL iterations



(b) 9 iterations with all methods

Figure 4.9: The average and std on 5-folds of the weighted f1 score on AGNews dataset. For each tested method (marked with CS) the bar on its left (marked with WS) represents the performance given D_L^{ws}. The bar line on the right is the BADGE baseline. Note figure a) on the right does not include Iterative T-CALR.

Looking more closely at the different cold start techniques, each one displays a unique performance trajectory. T-CALR excels among the other cold start techniques in the first four iterations, even if with high variability, but falls behind as the worst-performing technique afterwards. Interestingly, in the final iterations, the performance gap between T-CALR and the other techniques diminishes, making them comparably effective. Notably, T-CALR struggles with the *world* class, only achieving an F1 score above 0.8 in the last two iterations, matching BCS. On the other hand, it demonstrates impressive performance on the *sports* class, outperforming all other techniques from the second iteration.

The batches selected by the warm start technique (T-CALR (WS)) are consistently more effective, evident from the earliest iterations.

Iterative T-CALR performs the best (as per weighted F1 scores) among the proposed cold start methods until iteration 1 and matches BCS performance in the eighth and ninth iterations. However, it is surpassed by random sampling from the second iteration. Surprisingly, it even outperforms the simulated balance. The samples selected using the warm start technique (Iterative T-CALR (WS)) are more effective than those sampled by Iterative T-CALR from the first iteration, although it only outperforms random sampling in the initial iteration.

BCS overcomes T-CALR sampling at iteration 1 as per the weighted F1 bar plot and exhibits exceptional performance regarding the *world* class F1 bar plot compared to other cold start techniques.

Across the experiment, random sampling consistently achieves performance comparable to or better than BADGE sampling.

4.2.5 Analysis and Discussion

Initially, it must be highlighted that only 9 iterations were executed for the Iterative T-CALR method, as opposed to 19. This limitation was due to the time-consuming and resource-intensive nature of training the SBERT model through contrastive learning with setFit, particularly when the scale of training escalated.

In both datasets, it was observed that the T-CALR method struggled to match the efficiency of other techniques, especially with less represented classes. Techniques that effectively utilized label information were seen to demonstrate rapid progress. Although results on the Altilia dataset were in alignment with our initial hypothesis - illustrating that cold start techniques outperform warm start and random sampling in the initial iterations - the results on the AGNews dataset were surprising. An unexpectedly high performance was recorded for random sampling, even outpacing simulated balance sampling, suggesting that the selection of the balanced pools may not always guarantee higher performance.

Significant insights were gained from the performance contrast between T-CALR and Iterative T-CALR methods. The fixed sampling framework of the T-CALR method, which relied on initial unsupervised clustering, was identified as its major weakness. If the initial clustering was not efficient, the sampling via the T-CALR method invariably resulted in a biased outcome.

On the other hand, by continually refining its embeddings, the Iterative T-CALR method was found to enhance the effectiveness of the examples used in training the target model. This process improved the representation of underrepresented classes and was even found to outperform the balancing objective of BCS in the challenging Altilia dataset. It was thus implied that sampling of typical data based on information density is more beneficial than merely retrieving examples from certain classes without considering their information properties.

Through the application of statistical tests, it was confirmed that the Iterative T-CALR method outperforms the traditional T-CALR method. The alternative hypothesis pro-

posed was that T-CALR performs stochastically lower than Iterative T-CALR. In the case of the Altilia dataset, this hypothesis was substantiated from the iteration 1 onwards. The Mann-Whitney U rank test and Mood's median test yielded a p-value below 0.02, indicating a statistically significant difference favoring the Iterative T-CALR considering a p-value of 0.05.

However, the AGNews dataset presented a slightly different scenario. While the Iterative T-CALR still performed better than the T-CALR, the p-value remained above the 0.05 threshold until the seventh iteration. From iteration three onwards, the p-value fell below 0.05, affirming the statistical significance of Iterative T-CALR's superior performance. These results robustly validate the effectiveness of the Iterative T-CALR method, demonstrating its superior performance over the traditional T-CALR in varying dataset conditions.

Another interesting pattern observed from the bar plots is that when the D_L^{ws} begins yielding better results than D_L^{cs} it keeps doing so for the next iterations. This indicates how there is actually a break point where BADGE has enough data to provide useful uncertainty measures about the unlabeled data and is not just a random event. If the features of the break point could be identified the active learning process would be optimized by exploiting first the cold start techniques and then substituting it with a warm start one.

The primary research question was focused on determining the number of iterations (or the number of labeled examples) required for warm start techniques to outperform cold start techniques. In the Altilia dataset, it was found that the BADGE warm start method required between 5 and 6 iterations to surpass the performance of the cold start methods. However, in the AGNews dataset, BADGE was able to outperform the cold start methods from the very beginning, suggesting that the number of iterations or the size of the labeled pool is not the sole determinant of BADGE's better performance w.r.t. CS methods.

Despite the insights provided by this study, some limitations were recognized. Ideally, all 19 iterations should be tested for the Iterative T-CALR method to validate whether its performance advantage over other techniques is sustained. Due to the computational costs of BADGE on large datasets, complete text classification datasets could not be used in this study. Future work should aim to address this by using larger datasets with similar cross-validation settings to those used in literature. Furthermore, given that batch sizes were set to 100, the granularity of the experiment may hide patterns related to the performance of different methods.

4.2.6 Conclusions

In this experiment, several cold start techniques were tested, their performance was compared against random sampling and a warm start technique, BADGE. To test the hypothesis that balanced sampling would yield superior results, a completely balanced sampling method was also simulated. The experiment was conducted on two datasets from diverse domains - academic (English) and real-world business (Italian).

The results obtained from the two datasets were not completely aligned. In the case of the Altilia dataset, cold start methods outperformed both BADGE and random sampling in the initial iterations, and complete balance demonstrated the highest effectiveness. Contrarily, for the AGNews dataset, cold start techniques lagged behind BADGE from the

outset, and simulated balance could not even match the performance of random sampling.

However, noteworthy differences in the performance of different techniques were identified. Iterative T-CALR displayed a significant improvement over T-CALR, revealing that enhanced embeddings, even when the clustering and selection steps are unaltered, can lead to the effective selection of typical data.

Overall, the experiment demonstrated that even if the quantity of labeled data is limited, and the methods employed are naive, leveraging them can significantly enhance the effectiveness of active learning cycles. This was evidenced by the performance of BCS and Iterative T-CALR in particular.

For future work, conducting experiments with larger, standard datasets from literature would be beneficial, providing comparable results and the opportunity to evaluate other cold start techniques such as ALPS. This study is a step forward in understanding and refining cold start techniques, and it lays the groundwork for further exploration and development in this field.

Chapter 5

Conclusions

Research Questions and Insights

This research tackled the challenge of enhancing the active learning process by delving into the cold start phase, comparing various cold start techniques, and discerning the point at which warm start techniques surpass their cold start counterparts. The study was guided by two main research questions:

- Can cold start techniques furnish an initial labeled pool that speeds up subsequent active learning iterations? This inquiry is based on the notion that a well calibrated chosen initial labeled pool could assist warm start techniques in selecting more informative examples in the subsequent iterations, thus mitigating sampling bias.
- When do warm start techniques surpass the effectiveness of cold start ones? With an expanding pool of labeled samples, the reliability of uncertainty measures employed by warm start techniques is anticipated to improve. The pivotal question centers on identifying the point at which warm start techniques outperform their cold start counterparts.

The underlying hypothesis posited that while cold start techniques would excel in the early stages of the active learning process, they would be surpassed by warm start techniques as active learning progresses, due to their skilled application of uncertainty measures. However, the optimized selection of initial samples could yield significant enhancements to the overall process.

The study revealed significant findings and insights. In the initial experiment, which assessed the effectiveness of different cold start techniques in kickstarting active learning cycles using Altilia's dataset, the research sought to address the first research question. The results revealed that the presence of typical data, and notably, class balance in the initial set, brought significant influence over the effectiveness of active learning in its early stages. While their impact diminished as active learning advanced, the simulation of complete balance sampling emerged as the most fruitful approach across all active learning iterations. Although none of the proposed techniques achieved complete balance, these results established an upper limit and furnished valuable insights into the characteristics of an ideal initial dataset.

In the second experiment, conducted on datasets from different domains yet characterized by similar class distributions, the performance of the tested methods and baselines

did not align perfectly in both cases. As a result, a definitive answer to the second research question was not conclusively obtained. Nonetheless, the result curves unequivocally demonstrated that Iterative T-CALR outperformed T-CALR consistently throughout the active learning cycles. This underscores the significance of capitalizing on even a very small quantity of labeled examples and underscores the imperative for further exploration of methods designed to leverage limited labeled data.

This thesis represents a substantial step forward in understanding and advancing the application of cold start active learning techniques in text classification, offering valuable insights that have the potential to reshape the landscape of active learning methodologies.

Limitations

The research carried out in this thesis was subject to some limitations that impacted both the experimental setup and the resulting outcomes.

Firstly, while novel cold start methods were introduced in this work, they were not systematically compared with existing literature methods during the experiments. This absence of comparative analysis may introduce an element of uncertainty in the performance evaluations.

Additionally, certain hyperparameters associated with active learning cycles, such as budget (sampling size), were pre-determined and held constant. In the first experiment, the size of the initial labeled set was fixed at 396, which, while practical, does not guarantee optimality. In the second experiment, maintaining a fixed budget of 100 at each iteration may have limited the granularity of learning patterns that could be observed. Exploring a range of budget sizes could offer a more nuanced understanding of the active learning process.

A further limitation arose from the exclusive use of BADGE as the warm start technique. Employing a variety of warm start techniques in the initial experiment might have yielded diverse results, shedding light on the variability in outcomes originating from the same initial set constructed by the cold start technique.

A separate category of limitations pertains to the datasets employed for testing the proposed methods. In the first experiment, only one dataset was utilized, and the data provided by Altilia was downsized by 80% to accommodate time constraints. In the second experiment, further reduction was applied, involving the selection of specific classes from Altilia's data and a partial usage of the AGNews dataset. These constraints may have influenced the generalizability and robustness of the results.

Unanticipated challenges and constraints stemmed primarily from computational resource limitations and the time constraints inherent in a master's thesis. It is worth noting that comparing active learning techniques adds an additional computational overhead to the standard costs of deep learning training, effectively amplifying the efforts by the number of active learning iterations.

Future Work

Building upon the findings of this thesis, several promising avenues for future research emerge, each with the goal to enhance the field of cold start active learning for text classification of documents.

First, there is a need to further explore the optimal combination of cold start and warm start techniques. Delving into this relationship will allow for a more comprehensive understanding of how these techniques can cooperatively contribute to the effectiveness of the active learning process.
A critical area of investigation lies in the detailed analysis of active learning performance over successive cycles, particularly in response to varying initial labeled sets. By discerning the distinctive features that define an optimal initial labeled set, future research can refine and fine-tune the selection process, ultimately leading to more efficient and effective active learning strategies.

To strengthen the efficacy of cold start techniques, the development of a stopping function represents a critical next step. Such a function would provide a clear criterion for determining when to transition from cold start to warm start, optimizing the timing and resource allocation in the active learning process.
An even more ambitious undertaking involves the creation of a dynamic method that seamlessly integrates features from both cold and warm start approaches. This adaptive approach would adjust its behavior in response to evolving conditions, such as the progression of active learning cycles and the availability of labeled data. The ultimate aim would be to facilitate a smooth and gradual transition from a complete cold start paradigm to a conventional warm start methodology, optimizing performance at every stage of the process.

Furthermore, future research in the domain of cold start active learning should prioritize comprehensive comparative analyses with existing literature, leveraging diverse datasets from a range of domains. Standardized experimental approaches and datasets can provide a robust foundation for evaluating and benchmarking the effectiveness of cold start techniques, enhancing the reliability and reproducibility of findings across studies.

These potential directions for future research promise to deepen our understanding of cold start active learning and pave the way for more efficient and effective strategies in text classification of business documents.

Key Takeaways and Contributions

This thesis marks a significant stride in the realm of active learning for text classification. The central contributions lie in the introduction of novel methods harnessing cutting-edge Sentence Transformers. These methods not only facilitate cold start techniques but also capitalize on the limited pool of initial labeled examples in the early active learning cycles. Moreover, a robust experimental framework has been established to gauge the efficacy of cold start techniques, especially in their role as catalysts for standard active learning processes employing warm start methods.

In essence, this thesis emphasizes that substantial advancements are achieved one small step at a time, exemplified by the iterative nature of the active learning cycles. It under-

scores that in the pursuit of progress, we must recognize that big changes are often the culmination of incremental improvements.

Acknowledgments

I want to express my sincere appreciation to my university supervisor, Gwenn Englebienne, for his invaluable support and astute guidance right from the beginning of this research endeavor. His genuine interest in the subject matter, pragmatic technical feedback, and steadfast assistance have been indispensable throughout this journey. I would also like to express my appreciation to Elena Mocanu for her valuable input in the final stages of this thesis.

I also wish to extend my gratitude to the entire team at Altilia, where I conducted this research. Special thanks are due to Francesco Visalli, who warmly welcomed me into the company and provided crucial assistance in identifying pertinent business-related research gaps. At Altilia, I was fortunate to have the daily support of Prospero Papaleo and Antonio Lanza, who not only aided me in organizing experiments but also were pivotal in optimizing the distribution of computational resources, effectively managing the costs associated with my experiments across the team's infrastructure..

Their collective contributions have significantly enriched this thesis, and I am truly grateful for their assistance.

Bibliography

[1] Huggingface - wordpiece tokenization. https://huggingface.co/learn/nlp-course/chapter6/6, 2021. Accessed: 2023-06-10.

[2] Dana Angluin. Queries and concept learning. *Machine learning*, 2:319–342, 1988.

[3] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[4] Ricardo Barata, Miguel Leite, Ricardo Pacheco, Marco OP Sampaio, João Tiago Ascensão, and Pedro Bizarro. Active learning for imbalanced data under cold start. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–9, 2021.

[5] Eric B Baum and Kenneth Lang. Query learning can work poorly when a human oracle is used. In *International joint conference on neural networks*, volume 8, page 8. Beijing China, 1992.

[6] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[7] Etienne Brangbour, Pierrick Bruneau, Thomas Tamisier, and Stéphane Marchand-Maillet. Cold start active learning strategies in the context of imbalanced classification. *arXiv preprint arXiv:2201.10227*, 2022.

[8] Liangyu Chen, Yutong Bai, Siyu Huang, Yongyi Lu, Bihan Wen, Alan L Yuille, and Zongwei Zhou. Making your first choice: To address cold start problem in vision active learning. *arXiv preprint arXiv:2210.02442*, 2022.

[9] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[11] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15:201–221, 1994.

[12] Franck Dernoncourt and Ji Young Lee. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. *arXiv preprint arXiv:1710.06071*, 2017.

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

[14] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133, 1997.

[15] Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö Arık, Larry S Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 510–526. Springer, 2020.

[16] A Gasparetto, M Marcuzzo, A Zangari, and A Albarelli. A survey on text classification algorithms: from text to predictions. information 13, 83 (2022).

[17] Leonardo Di Perna Giulio Ravasio. Gilberto: An italian pretrained language model based on roberta. https://github.com/idb-ita/GilBERTo.git, 2019.

[18] Santiago González-Carvajal and Eduardo C Garrido-Merchán. Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*, 2020.

[19] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[20] Daniel Grießhaber, Johannes Maucher, and Ngoc Thang Vu. Fine-tuning bert for low-resource natural language understanding via active learning. *arXiv preprint arXiv:2012.02462*, 2020.

[21] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[22] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[23] Rong Hu, Brian Mac Namee, and Sarah Jane Delany. Off to a good start: Using clustering to select the initial training set in active learning. 2010.

[24] Qiuye Jin, Mingzhi Yuan, Shiman Li, Haoran Wang, Manning Wang, and Zhijian Song. Cold-start active learning for image classification. *Information Sciences*, 616:16–36, 2022.

[25] Jaeho Kang, Kwang Ryel Ryu, and Hyuk-Chul Kwon. Using cluster-based sampling to select initial training set for active learning in text classification. In *Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004. Proceedings 8*, pages 384–388. Springer, 2004.

[26] Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher D Manning. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. *arXiv preprint arXiv:2107.02331*, 2021.

[27] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[28] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32, 2019.

[29] Mona Köhler, Markus Eisenbach, and Horst-Michael Gross. Few-shot object detection: A comprehensive survey. *arXiv preprint arXiv:2112.11699*, 2021.

[30] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[31] David D Lewis. A sequential algorithm for training text classifiers: Corrigendum and additional data. In *Acm Sigir Forum*, volume 29, pages 13–19. ACM New York, NY, USA, 1995.

[32] Xiaoxu Li, Zhuo Sun, Jing-Hao Xue, and Zhanyu Ma. A concise review of recent few-shot meta-learning methods. *Neurocomputing*, 456:463–468, 2021.

[33] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

[34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[36] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL: http://www.aclweb.org/anthology/P11-1015.

[37] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[38] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de La Clergerie, Djamé Seddah, and Benoît Sagot. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*, 2019.

[39] Andrew McCallum, Kamal Nigam, et al. Employing em and pool-based active learning for text classification. In *ICML*, volume 98, pages 350–358. Citeseer, 1998.

[40] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[41] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[42] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6707–6717, 2020.

[43] Jeroen Ooms. *tesseract: Open Source OCR Engine*, 2023. https://docs.ropensci.org/tesseract/ (website) https://github.com/ropensci/tesseract (devel).

[44] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[45] A Pranklin. *Introduction to the Theory of Statistics*. 1974.

[46] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[47] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[48] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

[49] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

[50] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[51] Christopher Schröder, Andreas Niekler, and Martin Potthast. Uncertainty-based query strategies for active learning with transformers. *arXiv preprint arXiv:2107.05687*, 2021.

[52] Christopher Schröder, Andreas Niekler, and Martin Potthast. Uncertainty-based query strategies for active learning with transformers. *arXiv preprint arXiv:2107.05687*, 2021.

[53] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE, 2012.

[54] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[55] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[56] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[57] Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*, pages 1308–1318. PMLR, 2020.

[58] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[59] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D13-1170.

[60] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[61] Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training. *arXiv preprint arXiv:2103.11955*, 2021.

[62] Alaa Tharwat and Wolfram Schenck. A survey on active learning: State-of-the-art, practical challenges and research directions. *Mathematics*, 11(4):820, 2023.

[63] Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. Efficient few-shot learning without prompts. *arXiv preprint arXiv:2209.11055*, 2022.

[64] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[66] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016.

[67] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[68] Lukas Wertz, Jasmina Bogojeska, Katsiaryna Mirylenka, and Jonas Kuhn. Evaluating pre-trained sentence-bert with class embeddings in active learning for multi-label text classification. In *2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (AACL-IJCNLP), online, 20-23 November 2022*, pages 366–372. Association for Computational Linguistics, 2022.

[69] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[70] Mark Woodward and Chelsea Finn. Active one-shot learning. *arXiv preprint arXiv:1702.06559*, 2017.

[71] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020.

[72] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020.

[73] Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. Cold-start active learning through self-supervised language modeling. *arXiv preprint arXiv:2010.09535*, 2020.

[74] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

[75] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[76] Zi-Ke Zhang, Chuang Liu, Yi-Cheng Zhang, and Tao Zhou. Solving the cold-start problem in recommender systems with social tags. *Europhysics Letters*, 92(2):28002, 2010.

[77] Fedor Zhdanov. Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*, 2019.

[78] Zongwei Zhou, Jae Shin, Lei Zhang, Suryakanth Gurudu, Michael Gotway, and Jianming Liang. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7340–7351, 2017.

[79] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

Appendix A

Appendix

A.1 Altilia dataset

A.1.1 Documents statistics

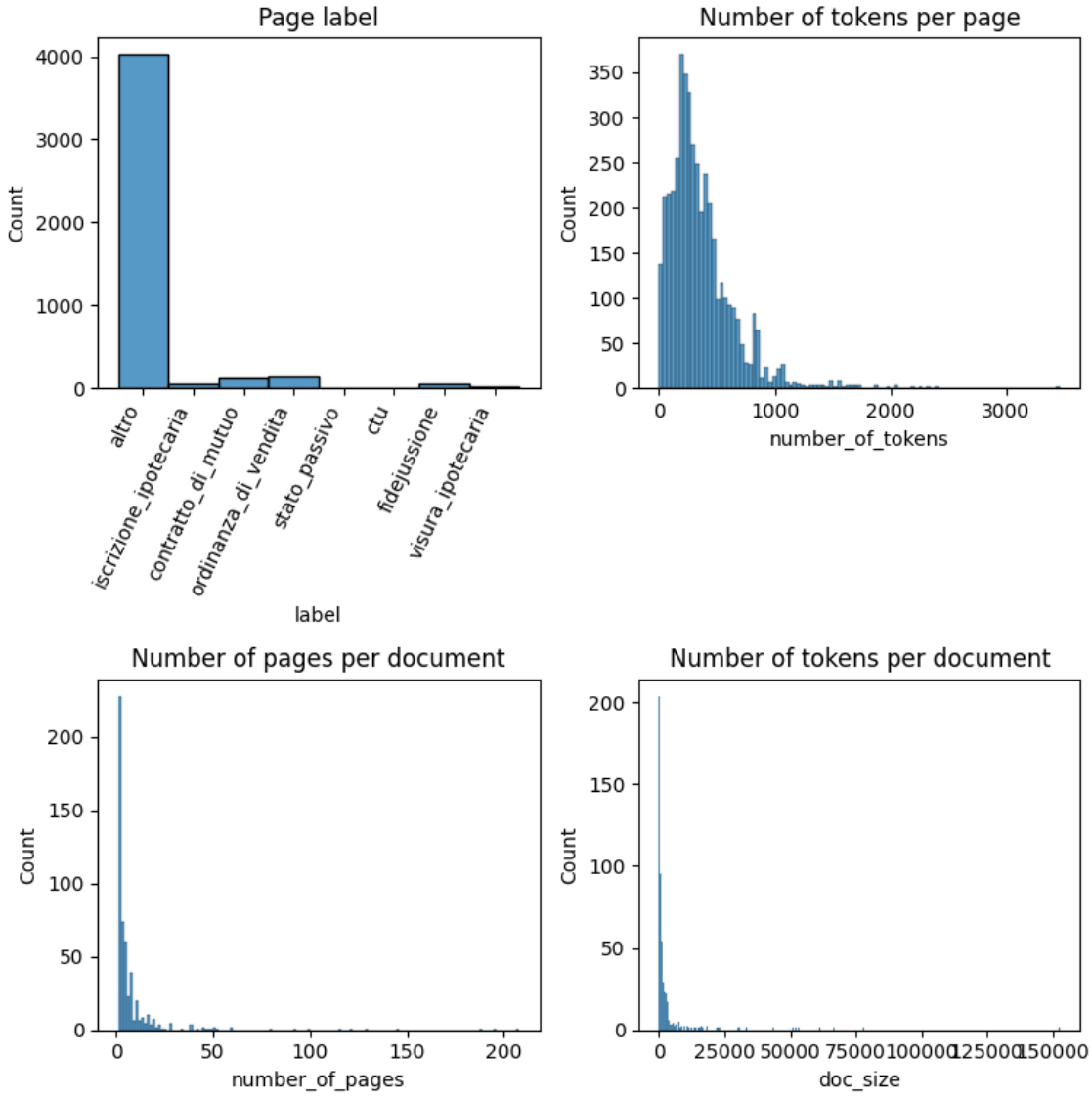


Figure A.1: *Altro* documents statistics

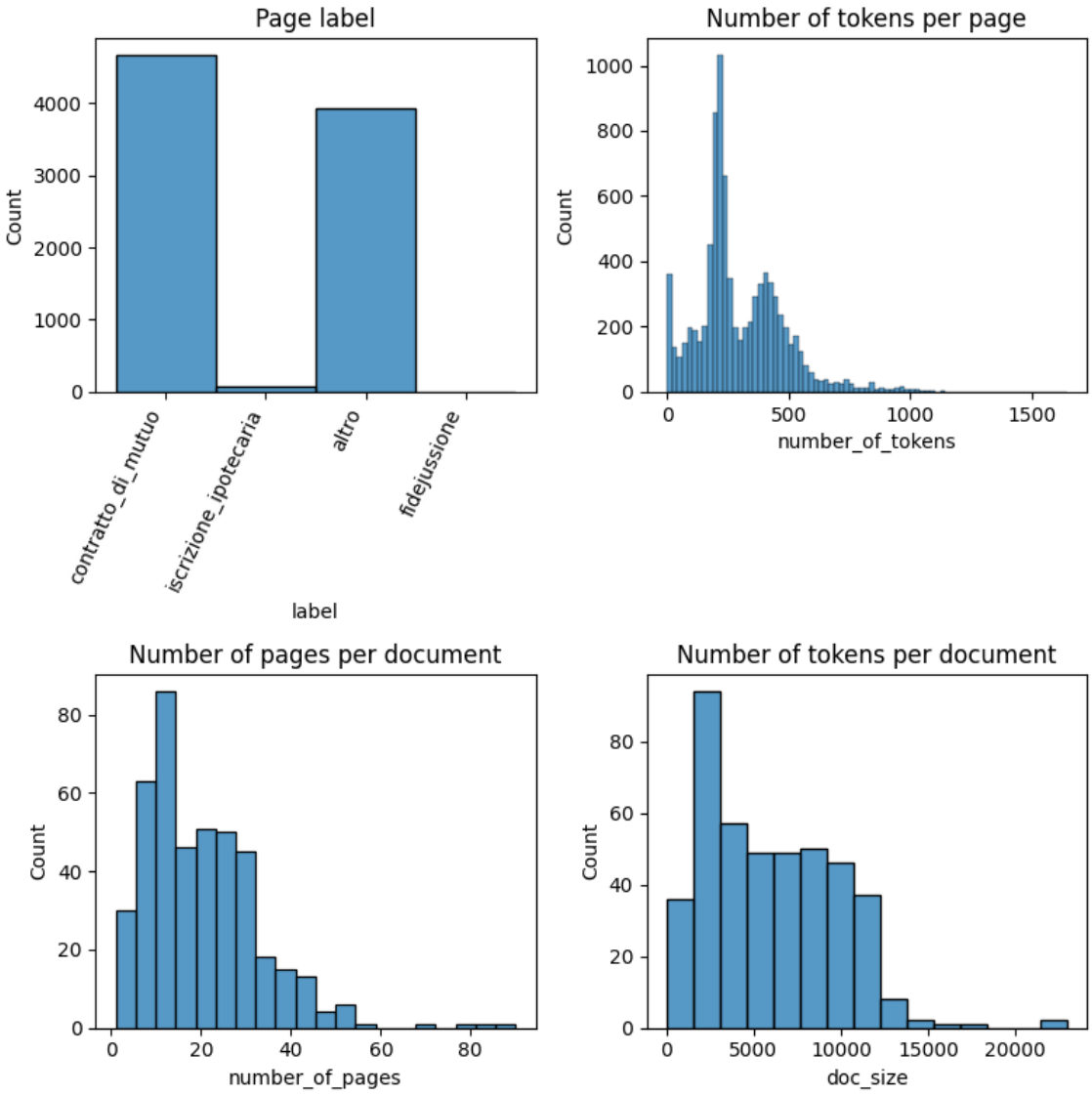


Figure A.2: *Contratto di mutuo* documents statistics

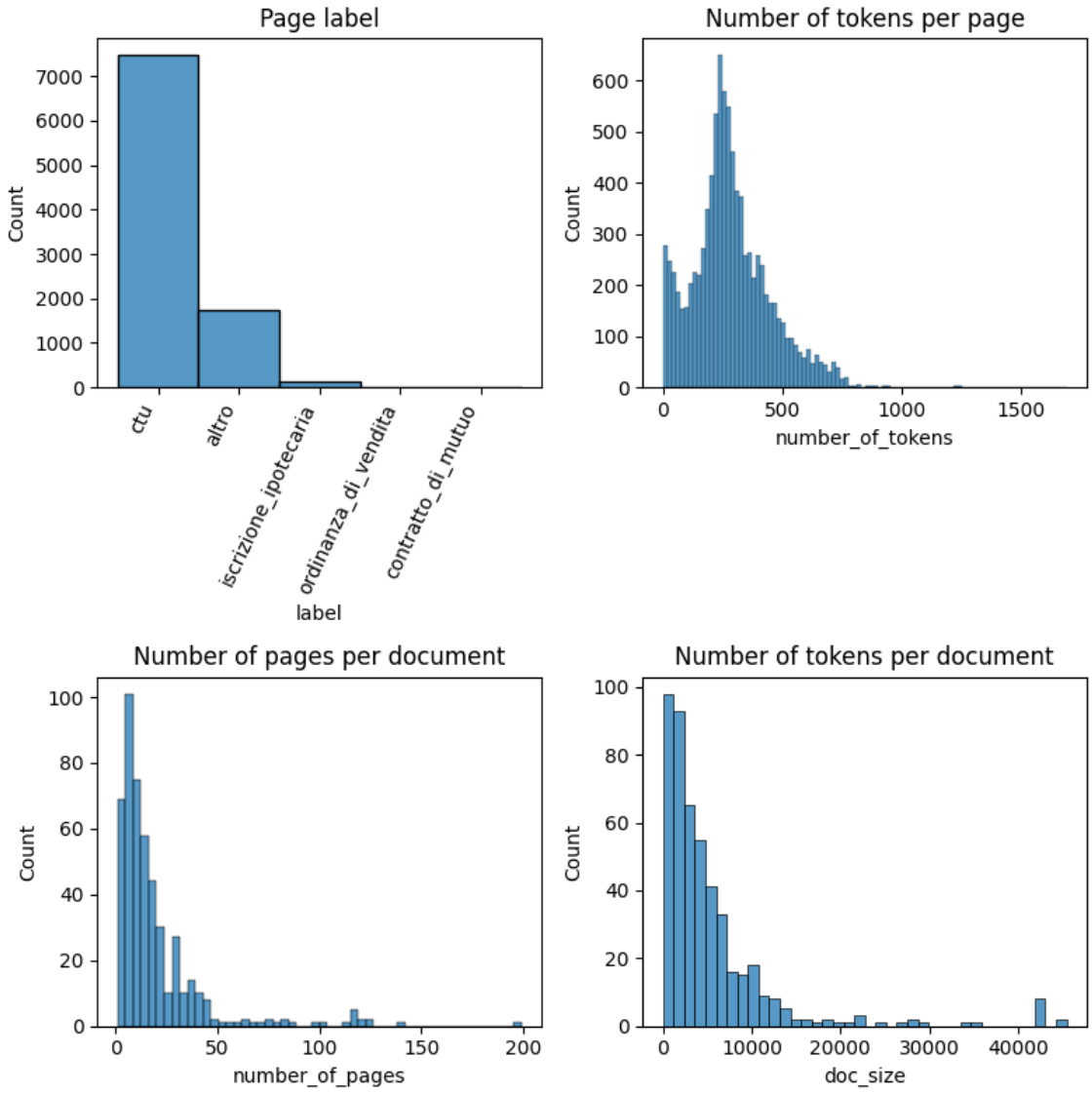


Figure A.3: *CTU* documents statistics

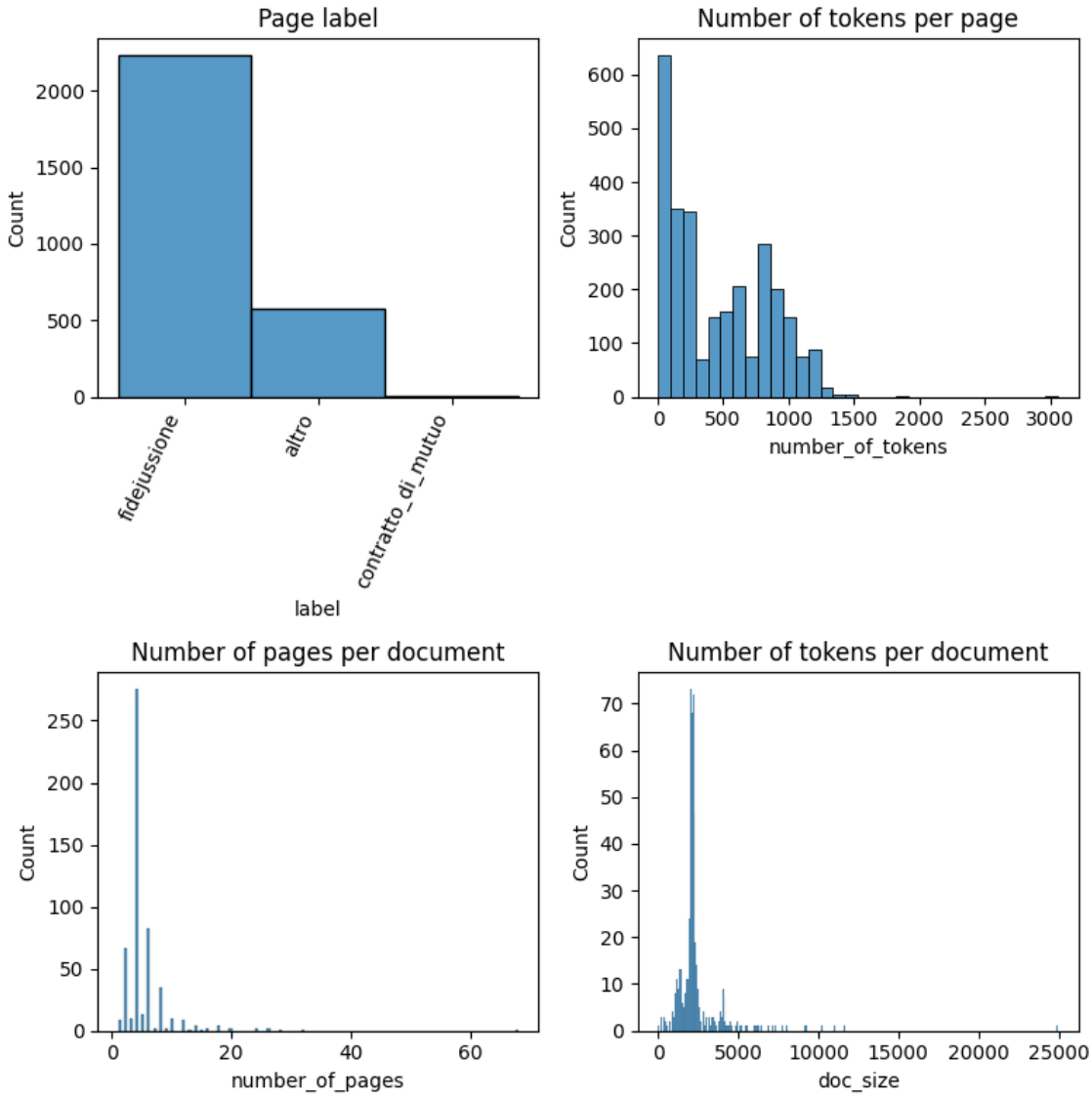


Figure A.4: *Fidejussione* documents statistics

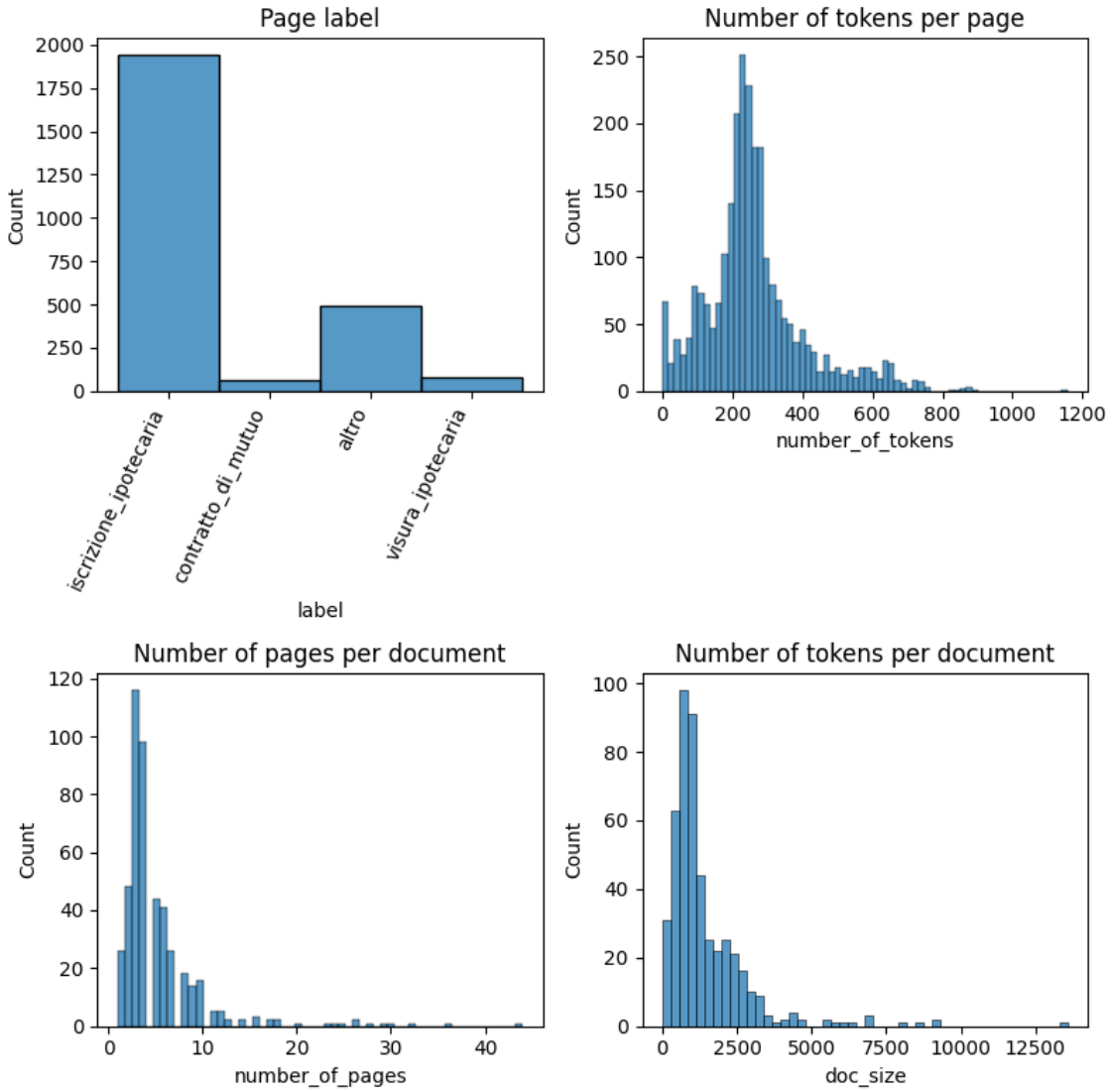


Figure A.5: *Iscrizione ipotecaria* documents statistics

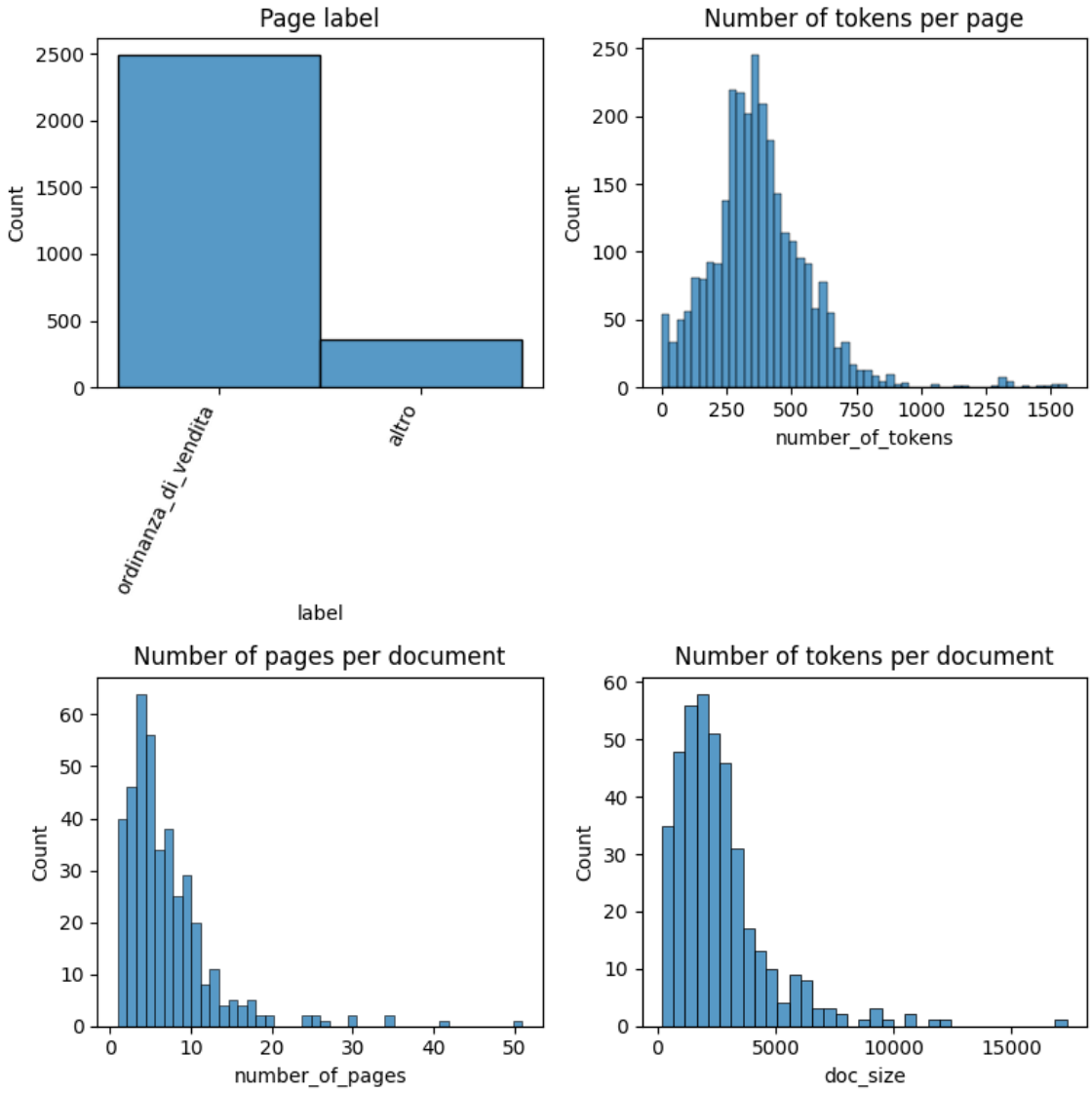


Figure A.6: *Ordinanza di vendita* documents statistics

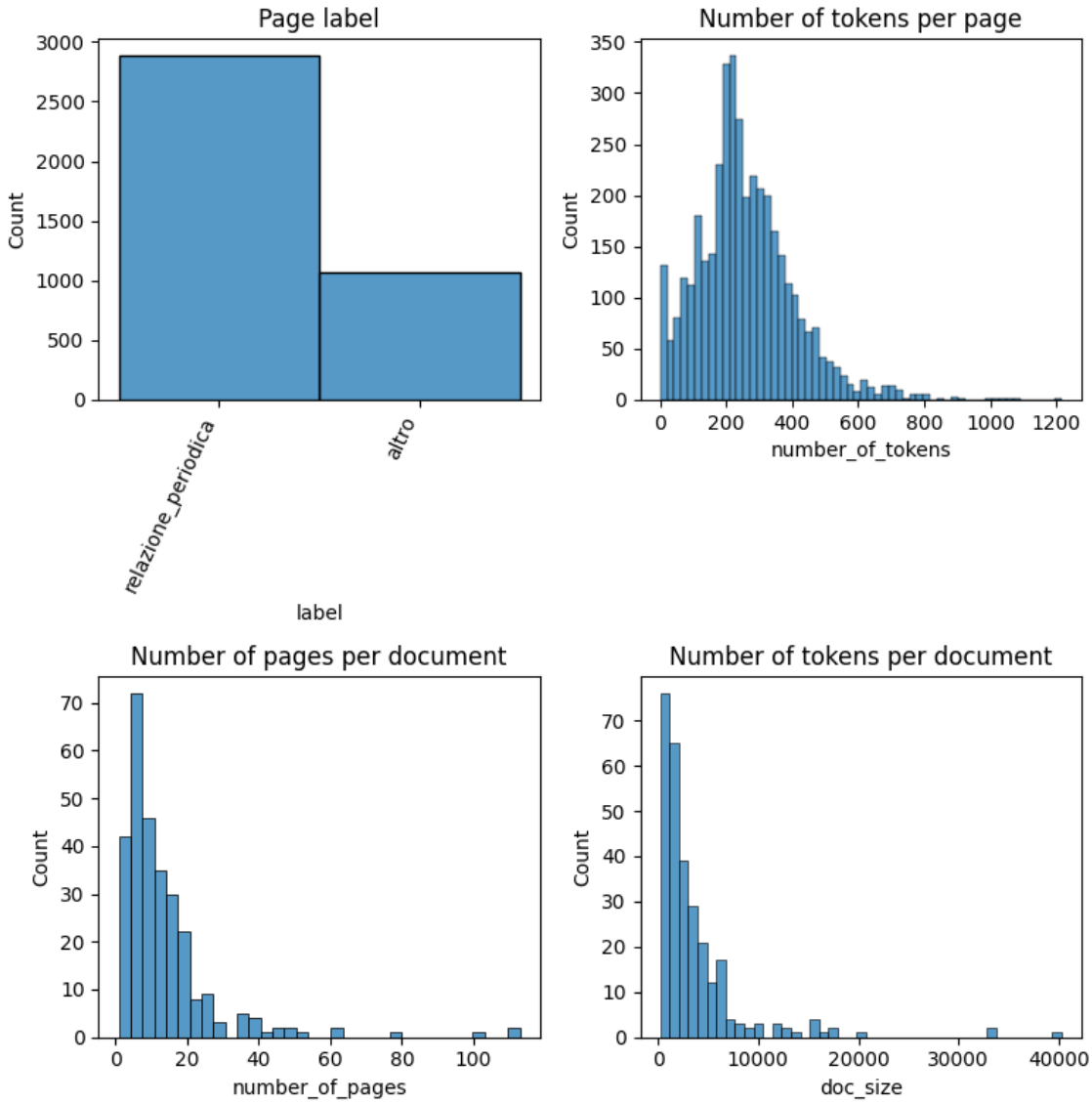


Figure A.7: *Relazione periodica* documents statistics

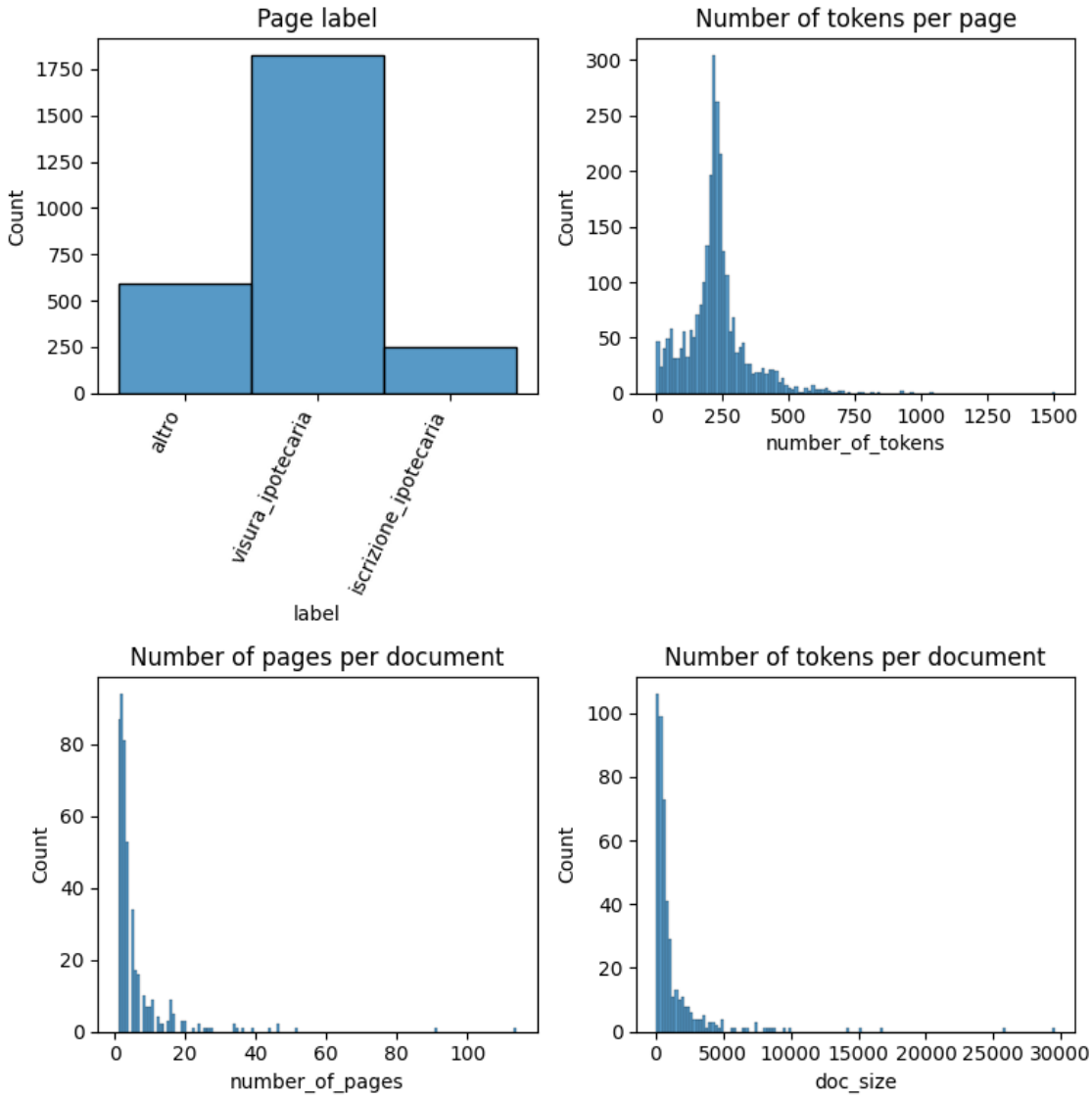


Figure A.8: *Visura ipotecaria* documents statistics

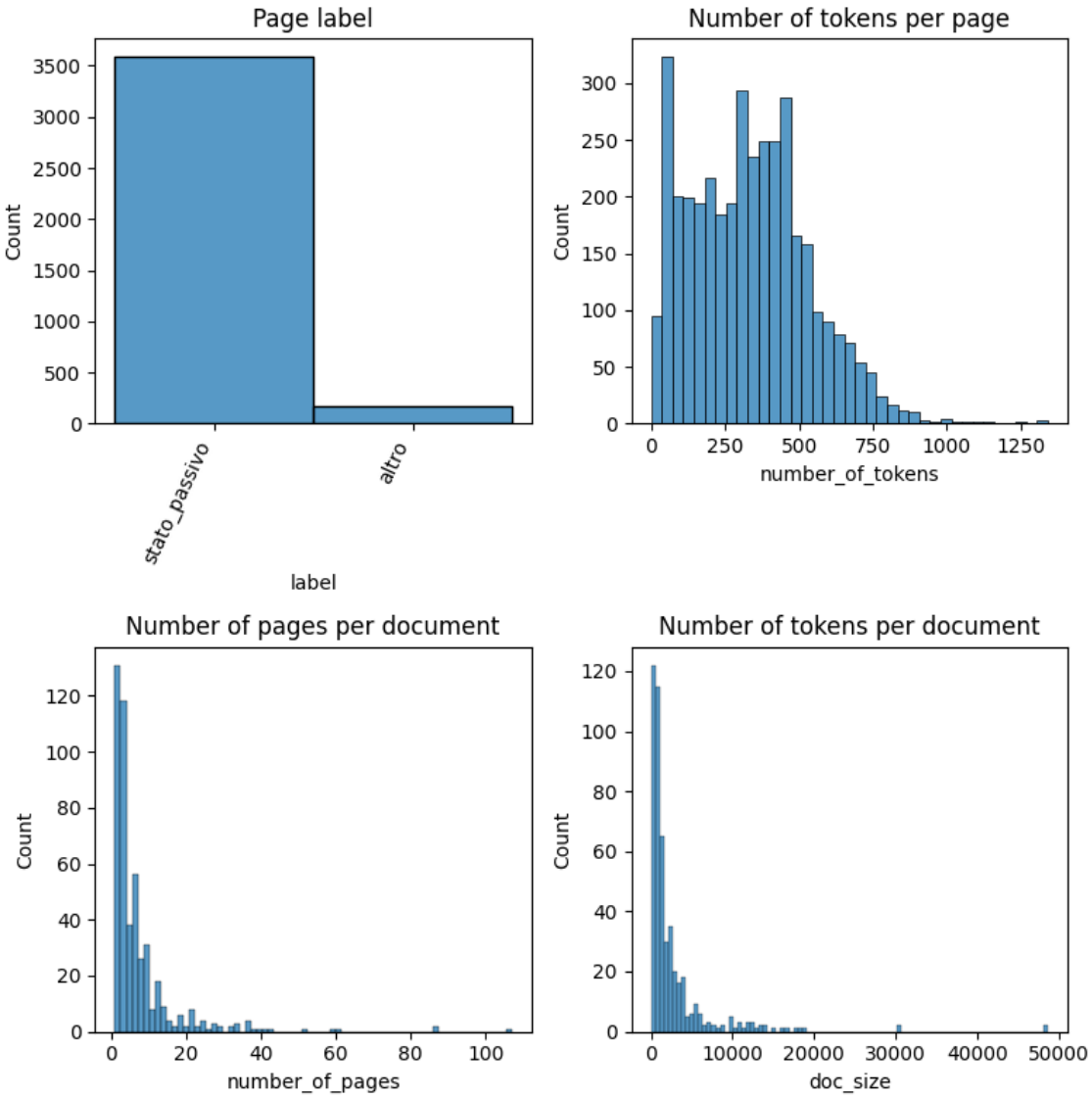
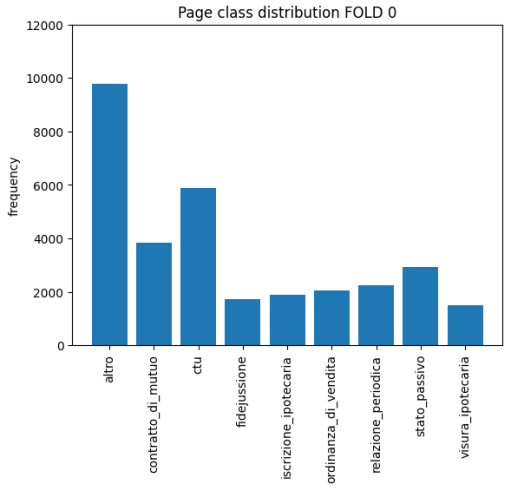
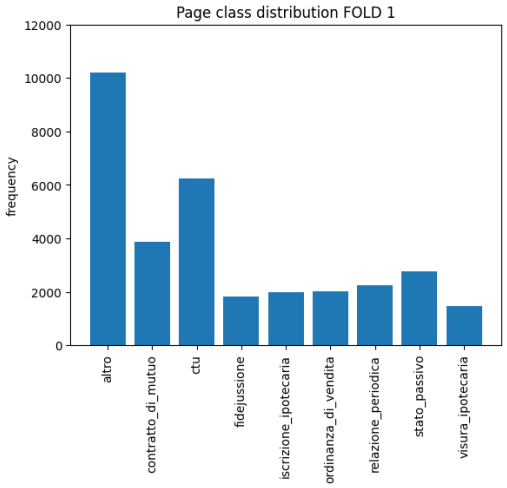


Figure A.9: *Stato passivo* documents statistics

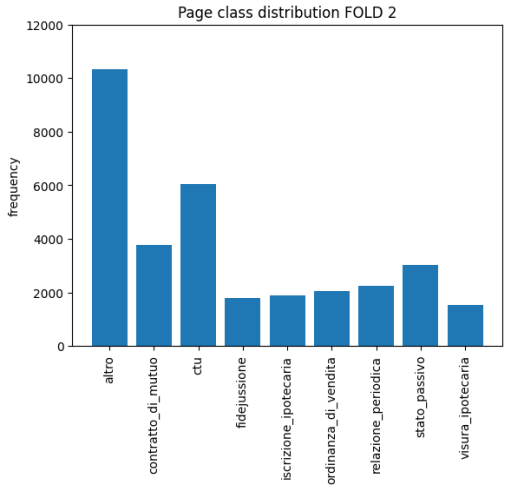
A.1.2 Folds statistics



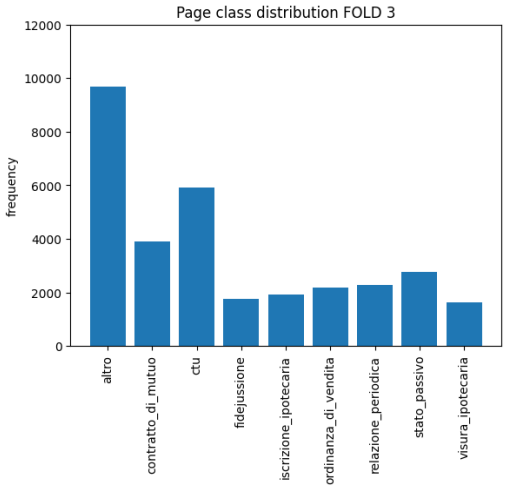
(a) Fold 0 pages' class distribution



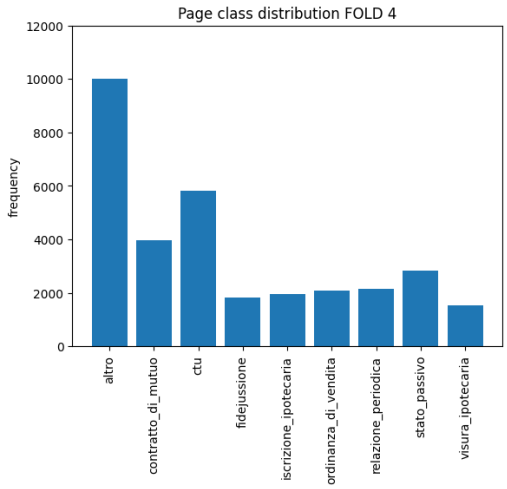
(b) Fold 1 pages' class distribution



(c) Fold 2 pages' class distribution

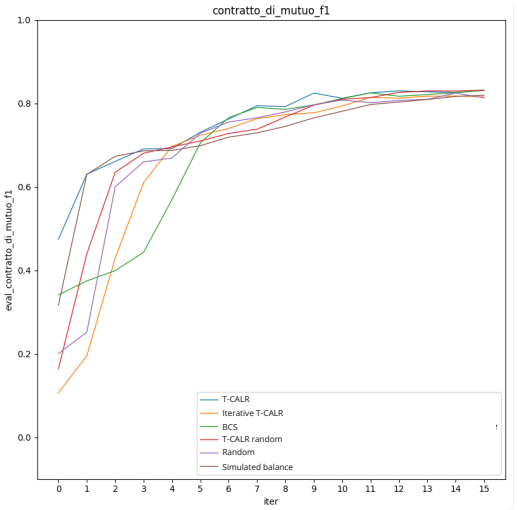


(d) Fold 3 pages' class distribution

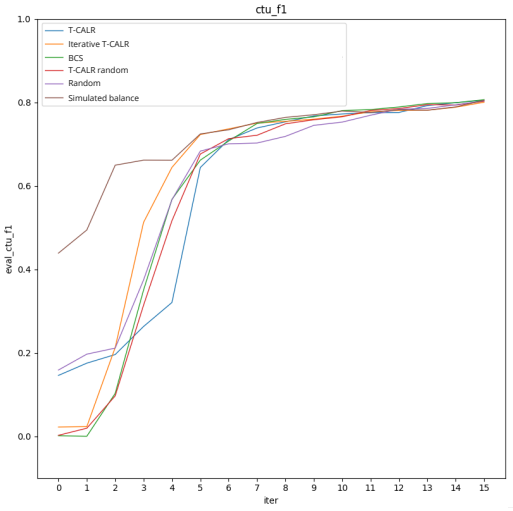


(e) Fold 4 pages' class distribution

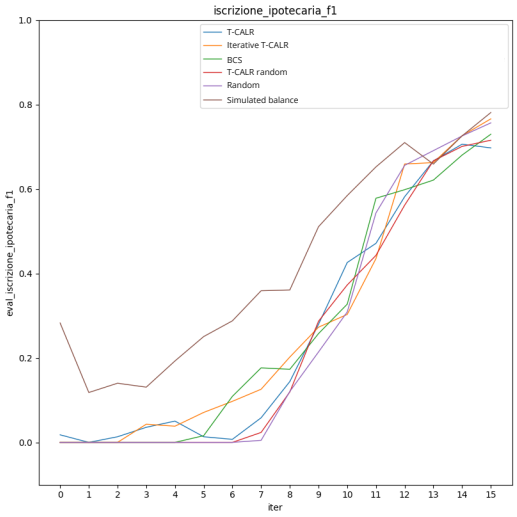
A.2 Experiment 1



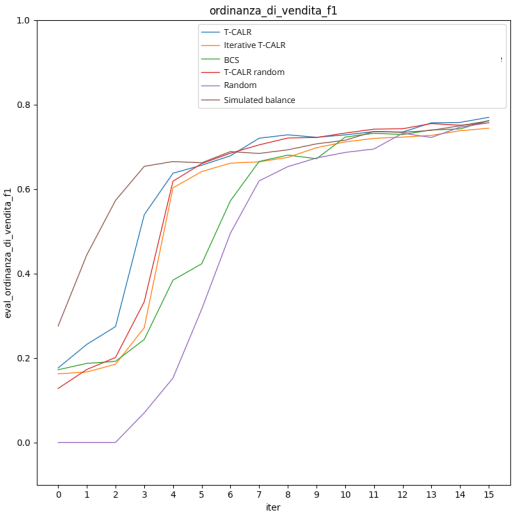
(a) Averaged on 3-folds f1 score of class *contratto di mutuo* for each active learning cycle



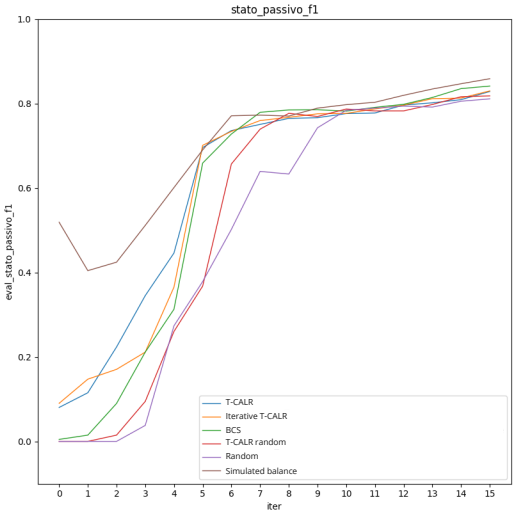
(b) Averaged on 3-folds f1 score of class *CTU* for each active learning cycle



(c) Averaged on 3-folds f1 score of class *iscrizione ipotecaria* for each active learning cycle



(d) Averaged on 3-folds f1 score of class *ordinanza di vendita* for each active learning cycle



(e) Averaged on 3-folds f1 score of class *stato passivo* for each active learning cycle

Figure A.11: Class f1 metrics across active learning cycles

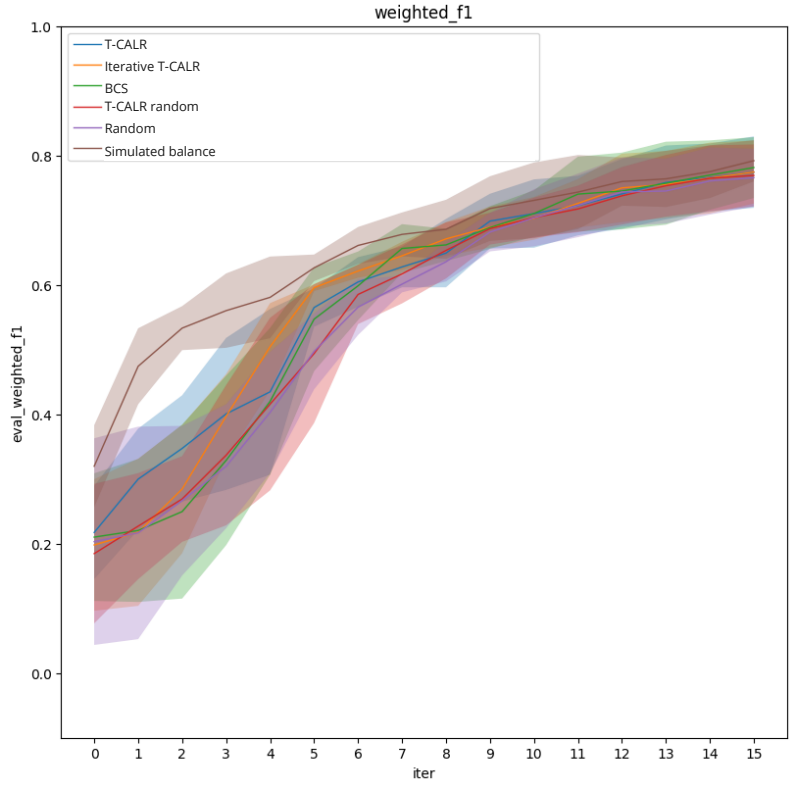
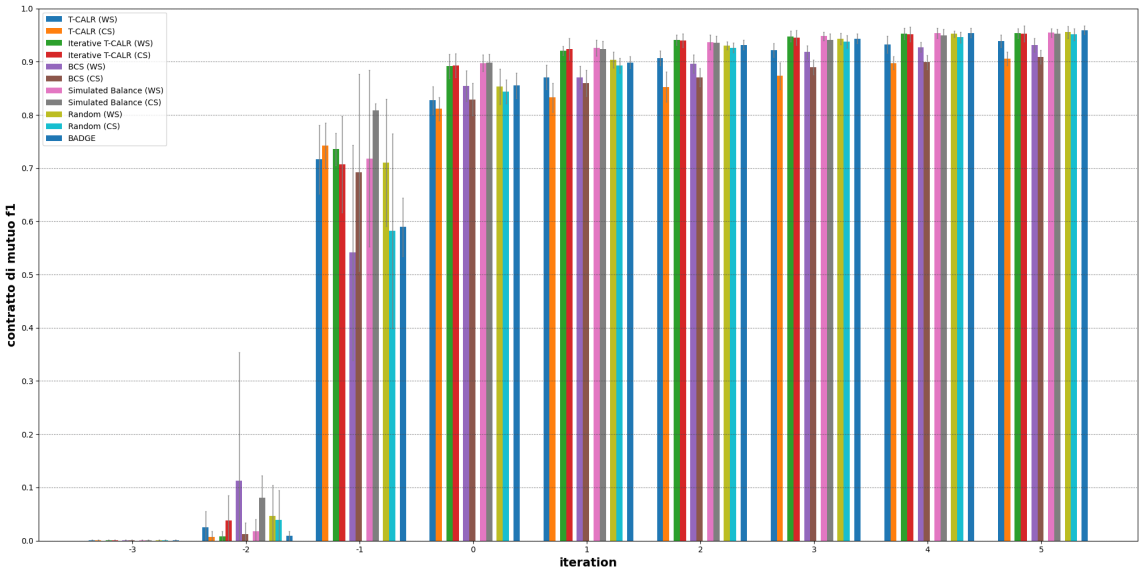


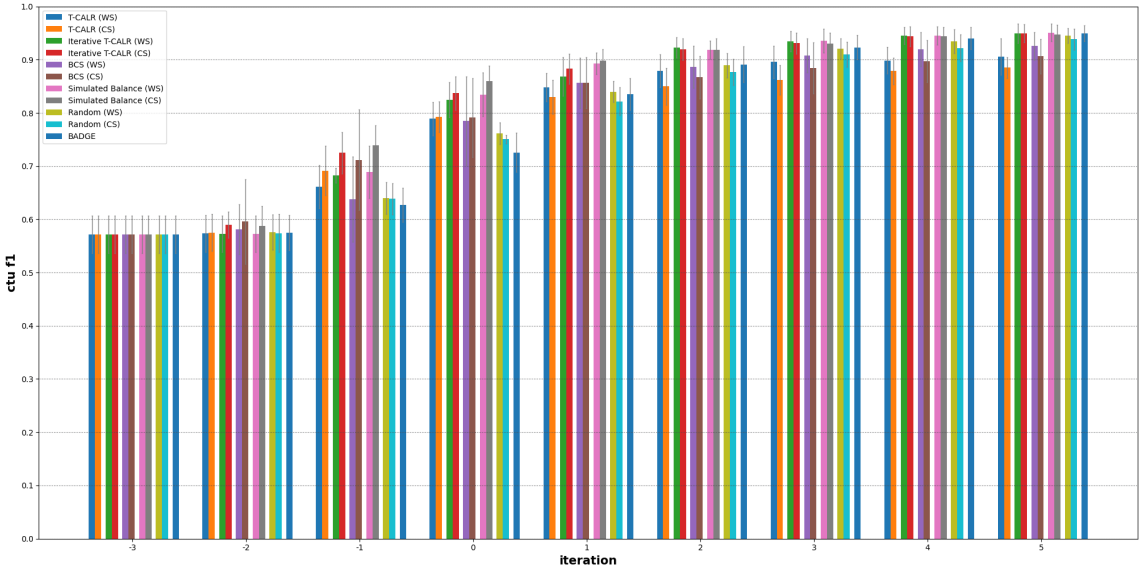
Figure A.12: Averaged on 3-folds weighted f1 score for each active learning cycle with std

A.3 Experiment 2

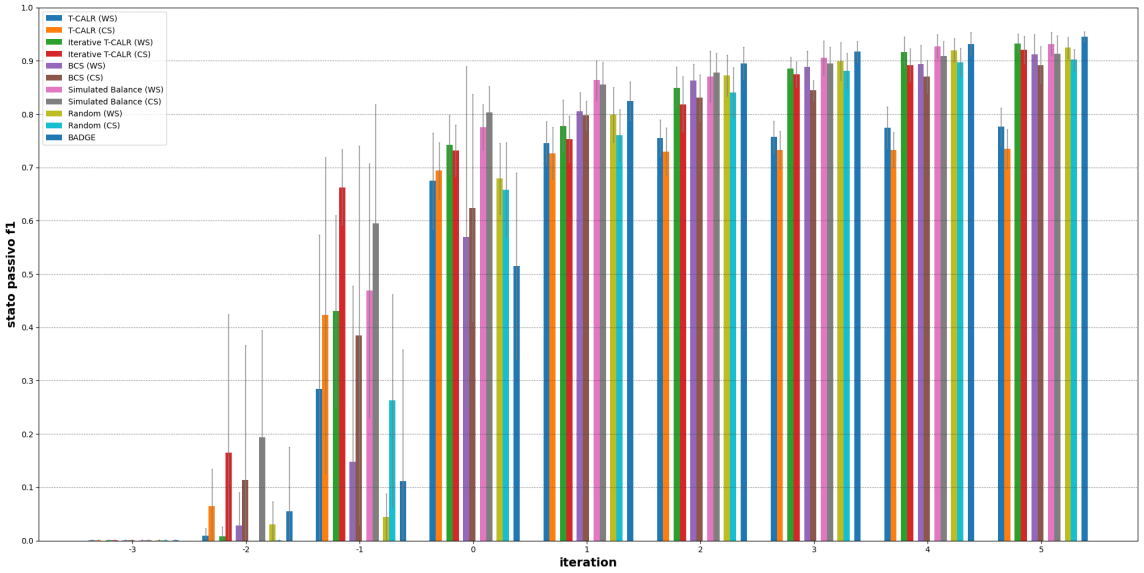
A.3.1 Results on Altilia dataset



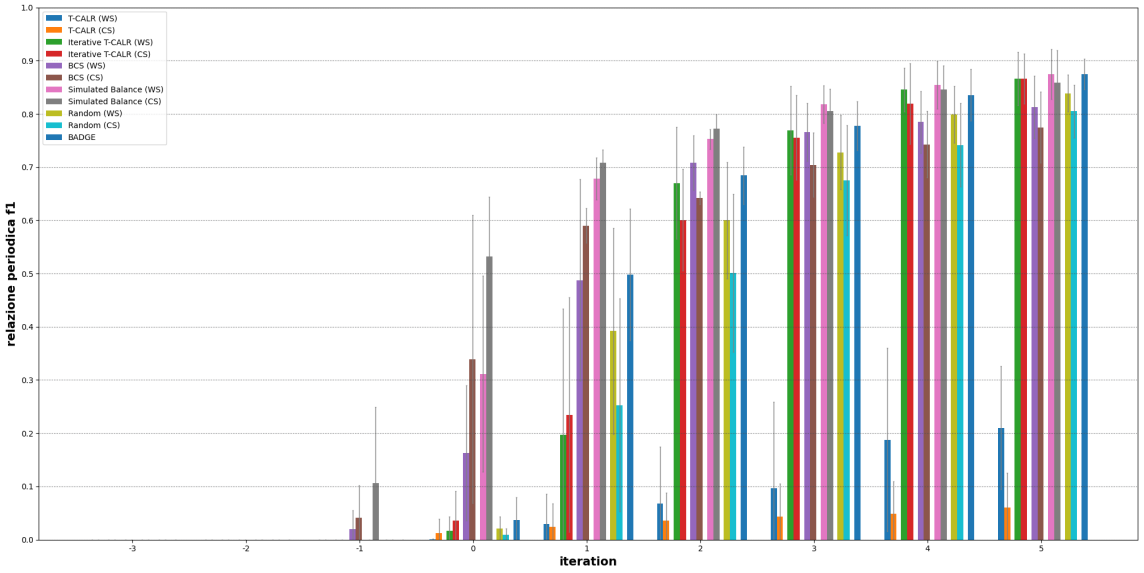
(a) Average on 5-folds, f1 score of class *contratto di mutuo* for each active learning cycle



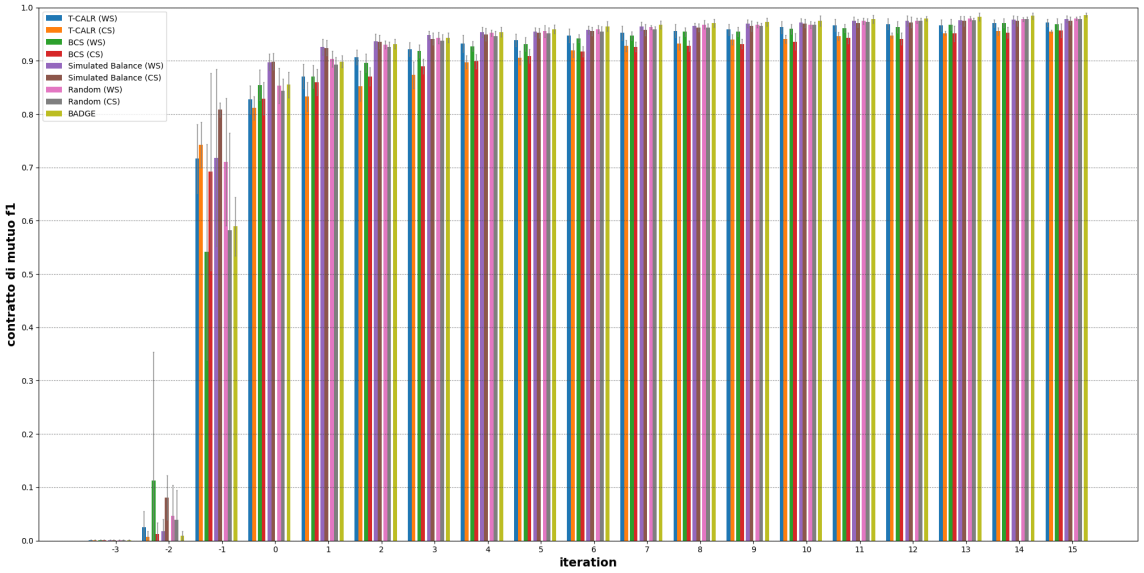
(b) Average on 5-folds, f1 score of class *CTU* for each active learning cycle



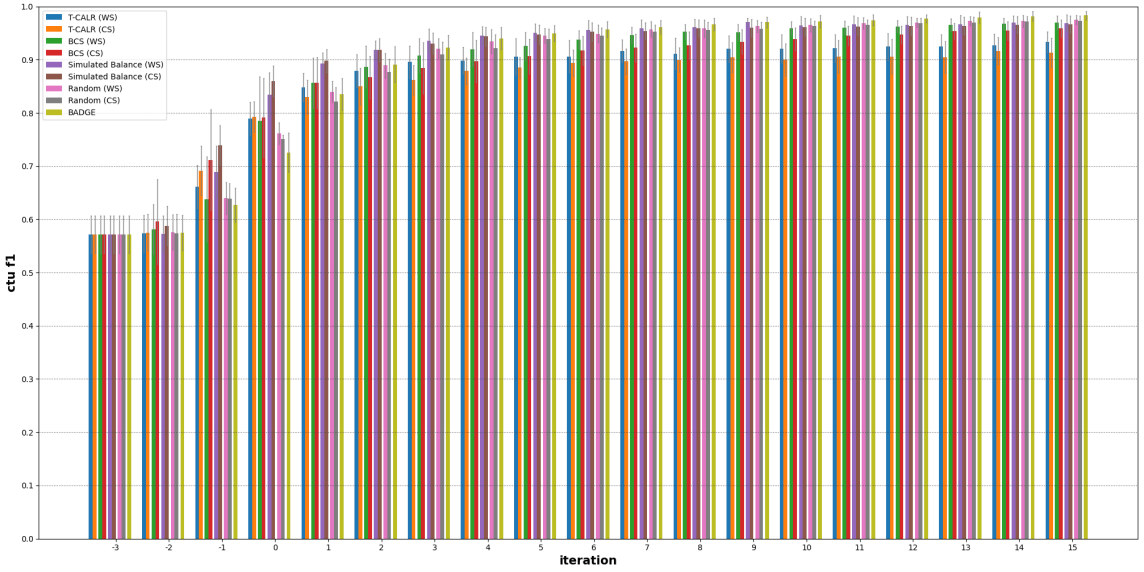
(c) Average on 5-folds, f1 score of class *stato passivo* for each active learning cycle



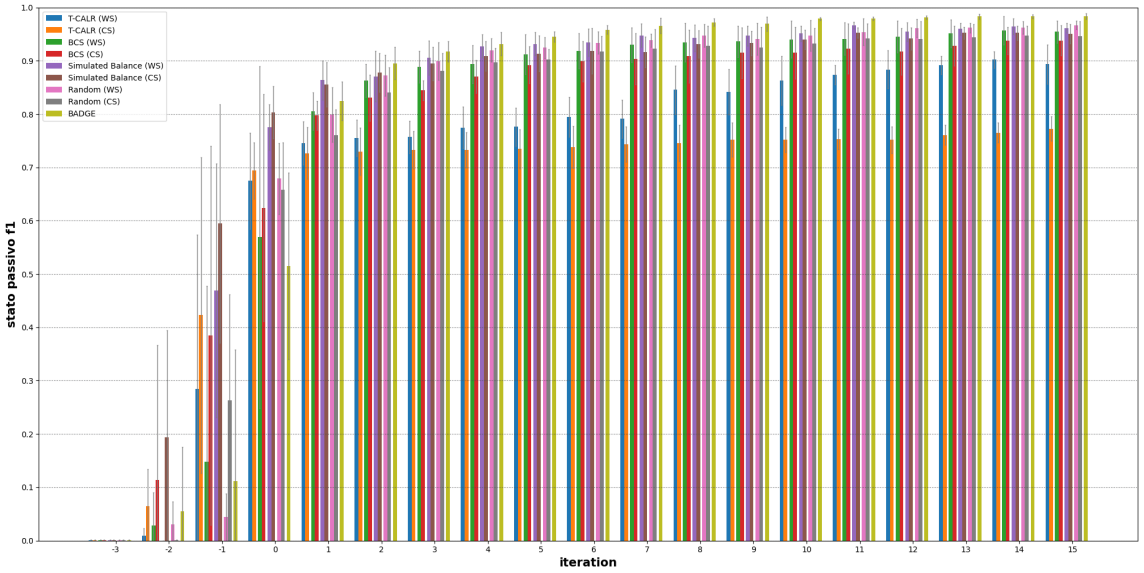
(d) Average on 5-folds, f1 score of class *relazione periodica* for each active learning cycle



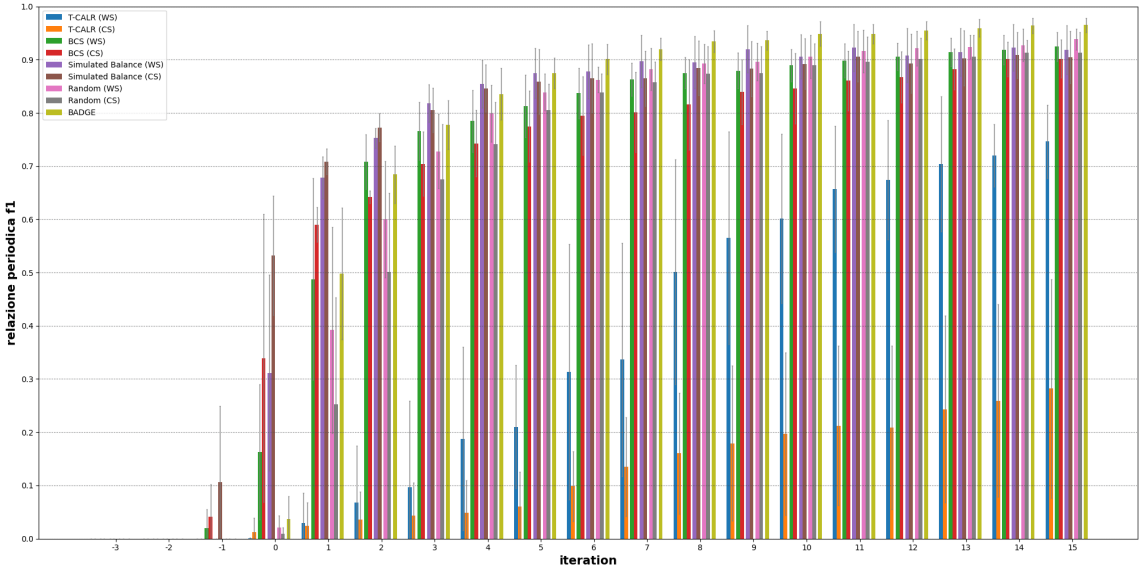
(a) Average on 5-folds, f1 score of class *contratto di mutuo* for each active learning cycle



(b) Average on 5-folds, f1 score of class *CTU* for each active learning cycle

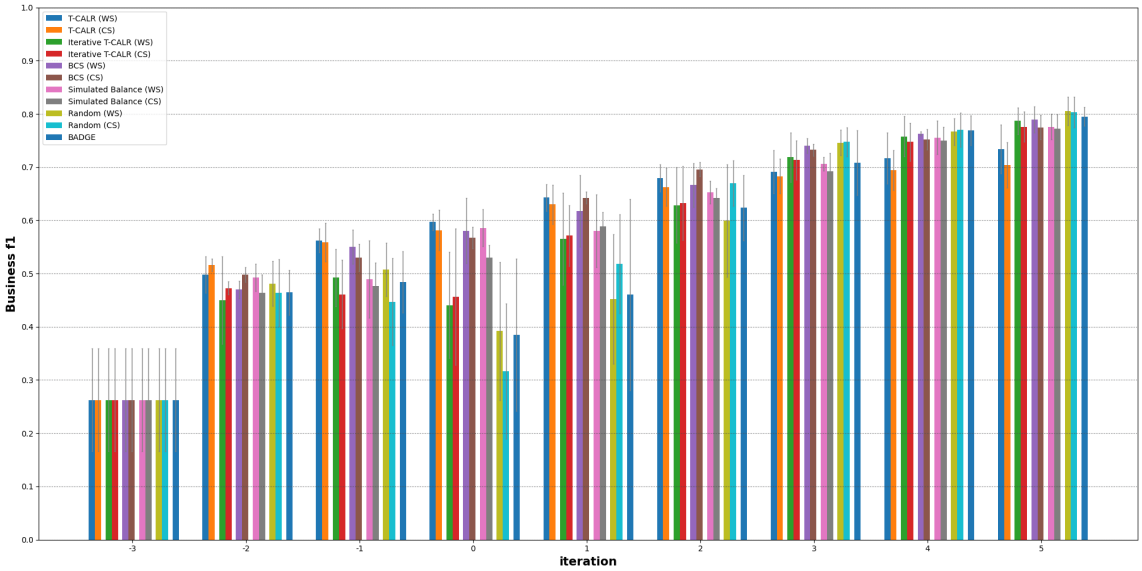


(c) Average on 5-folds, f1 score of class *stato passivo* for each active learning cycle

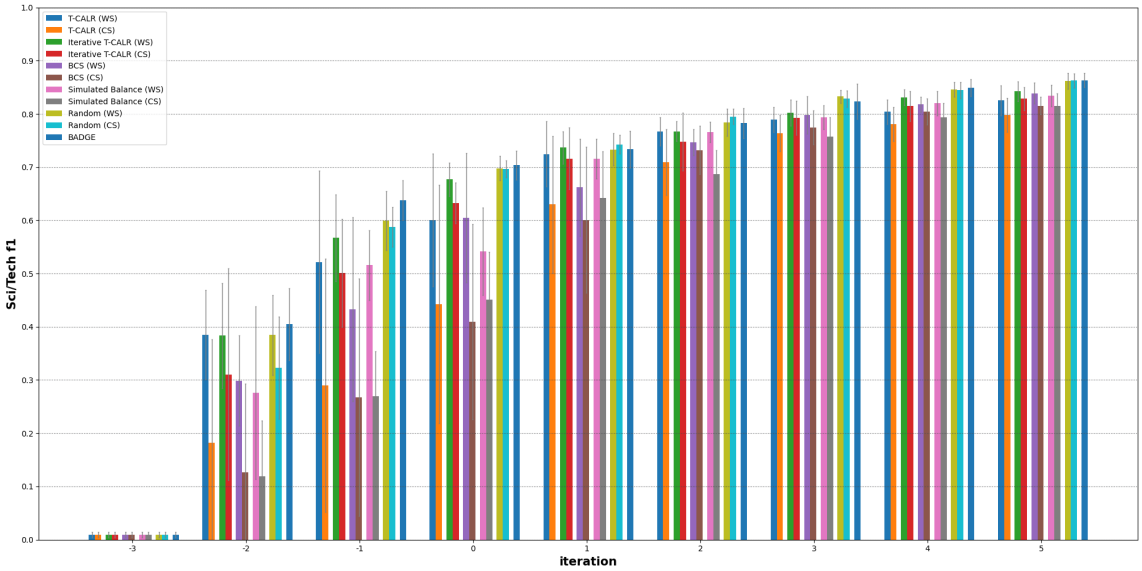


(d) Average on 5-folds, f1 score of class *relazione periodica* for each active learning cycle

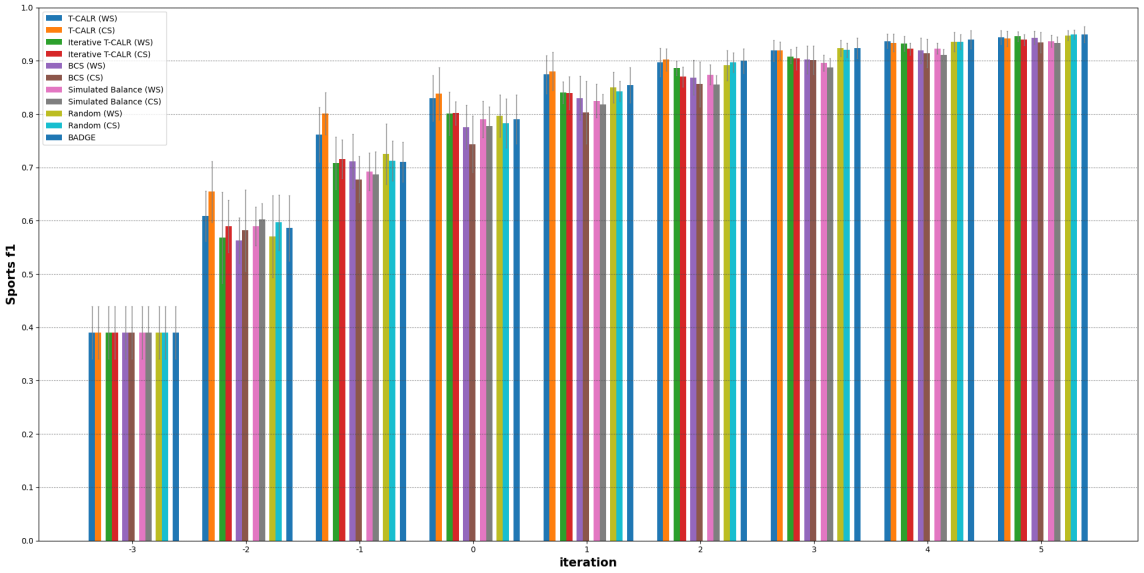
A.3.2 Results on AGNews dataset



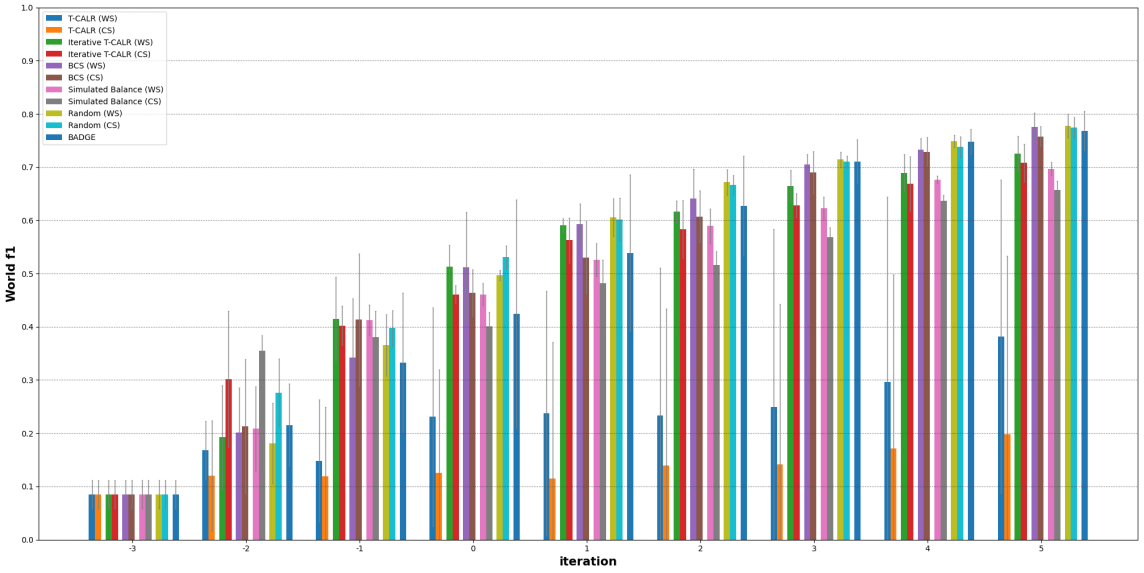
(a) Average on 5-folds, f1 score of class *business* for each active learning cycle



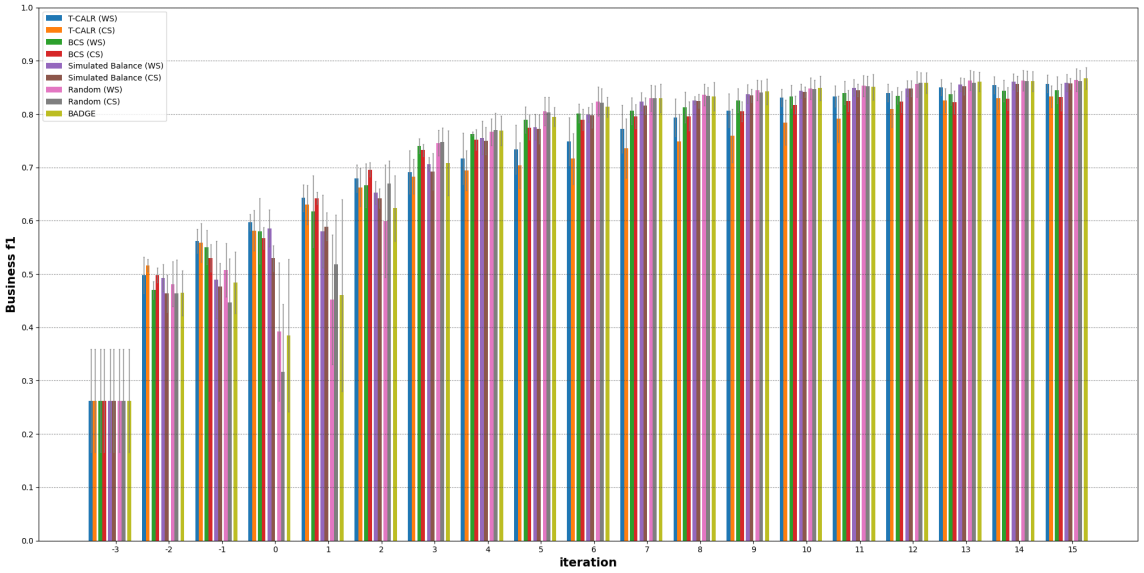
(b) Average on 5-folds, f1 score of class *sci/tech* for each active learning cycle



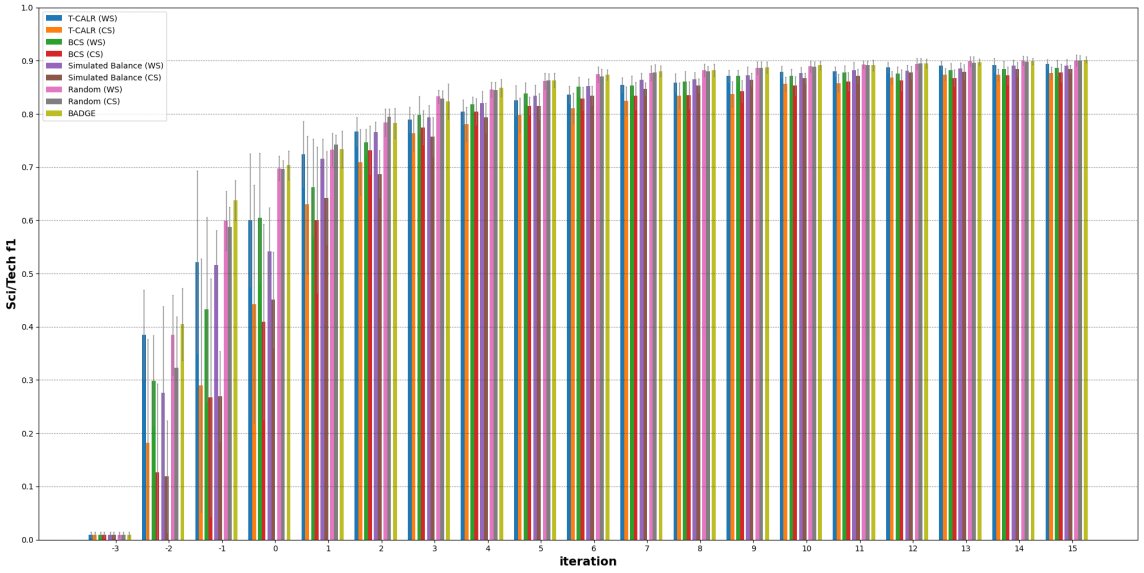
(c) Average on 5-folds, f1 score of class *sports* for each active learning cycle



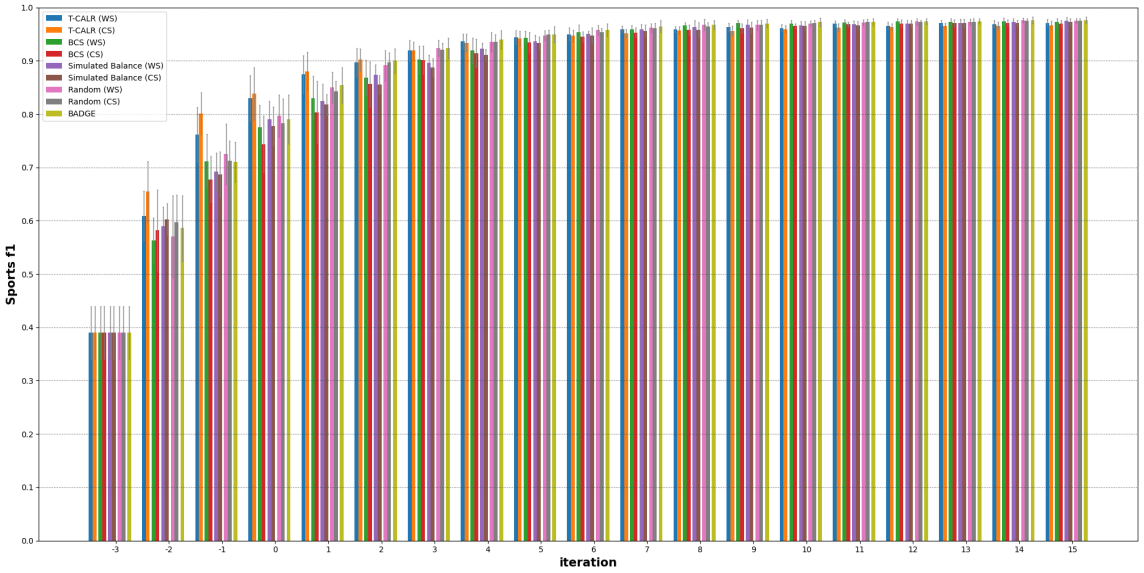
(d) Average on 5-folds, f1 score of class *world* for each active learning cycle



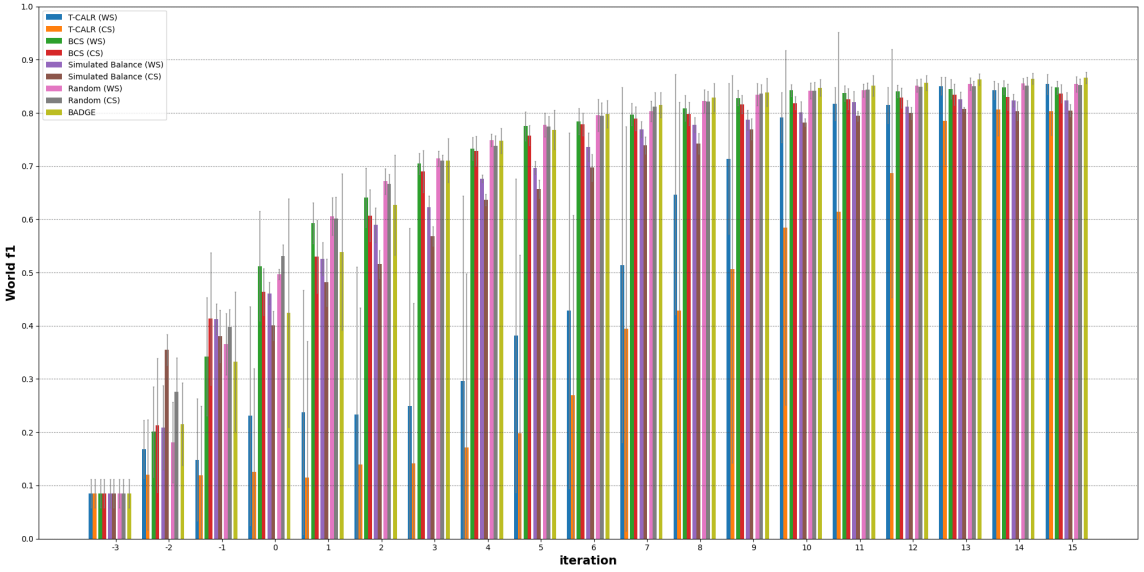
(a) Average on 5-folds, f1 score of class *business* for each active learning cycle



(b) Average on 5-folds, f1 score of class *sci/tech* for each active learning cycle



(c) Average on 5-folds, f1 score of class *sports* for each active learning cycle



(d) Average on 5-folds, f1 score of class *world* for each active learning cycle