UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Washington Luiz Miranda da Cunha

A Comprehensive Exploitation of Instance Selection Methods for Automatic Text Classification

Belo Horizonte
2024

Washington Luiz Miranda da Cunha

A Comprehensive Exploitation of Instance Selection Methods for Automatic Text Classification

Final Version

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Marcos André Gonçalves
Co-Advisor: Leonardo Chaves Dutra da Rocha

Belo Horizonte
2024

Cunha, Washington Luiz Miranda da.

C972c A Comprehensive exploitation of instance selection methods For automatic text classification [recurso eletrônico] / Washington Luiz Miranda da Cunha. – 2024.
1 recurso online (151 f. il, color.) : pdf.

Orientador: Marcos André Gonçalves.
Coorientador: Leonardo Chaves Dutra da Rocha.
Tese (Doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciências da Computação.
Referências: f.122-137

1. Computação – Teses. 2. Aprendizado do computador – Teses. 3. Classificação (Computadores) – Teses.
4. Processamento de linguagem natural – Teses. 5. Seleção de Instâncias. – Teses. I. Gonçalves, Marcos André. II. Rocha, Leonardo Chaves Dutra. III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Computação. IV.Título.

CDU 519.6*82.10(043)

Ficha catalográfica elaborada pela bibliotecária Irenquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

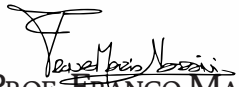
A Comprehensive Exploitation of Instance Selection Methods for Automatic Text Classification

WASHINGTON LUIZ MIRANDA DA CUNHA

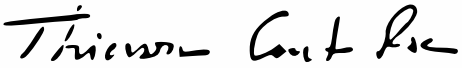
Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. MARCOS ANDRÉ GONÇALVES - Orientador
Departamento de Ciência da Computação - UFMG

PROF. LEONARDO CHAVES DUTRA DA ROCHA - Coorientador
Departamento de Ciência da Computação - UFSJ



PROF. FRANCO MARIA NARDINI
Istituto di Scienza e Tecnologie dell'Informazione Alessandro Faedo
Consiglio Nazionale delle Ricerche - CNR- Pisa



PROF. THIERSON COUTO ROSA
Instituto de Informática - UFG

PROF. RODRYGO LUIS TEODORO SANTOS
Departamento de Ciência da Computação - UFMG

PROF. ANISIO MENDES LACERDA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 26 de agosto de 2024.

This work is dedicated to all essential people in my life.

Acknowledgments

I'd like to express my gratitude to God for giving me health and protection all the time.

I'd like to express my deepest gratitude and appreciation to my wife, Adriana. Her unwavering support, boundless love, affection, dedication, and faith have been the cornerstone of this project and my life. I'm sure none of this would have been possible without her. Moreover, as this dissertation was defended on her birthday, I'd also like to wish her a Happy Birthday, my love.

To my parents, Washington and Ivone, for their unwavering encouragement, the invaluable education they provided, and the boundless love they showered on me.

To my brothers and friends who stayed with me during the whole Ph.D. journey. Although I was often physically absent, I carry each of you in my heart. I love you all!

I would like to express my heartfelt appreciation to Mr. Adilson and Mrs. Terezinha for their invaluable advice and guidance. Their thoughtful care and insightful guidance have been instrumental in my personal and professional growth.

To Dona Maria Lucia, Stella Cristina, and Aurora Maria for all the excitement and affection poured into me. Your support was crucial for my life and happiness.

I'd like to thank Professor Marcos and Professor Leonardo for all the knowledge, mentoring, and advice during my career, which were essential for achieving these results.

Finally, I also want to extend my gratitude to Professor Fabrizio Sebastiani and his incredible group for hosting me as a Researcher Visitor at the ISTI-CNR. It was an amazing time!! So, *grazie mille, ragazzi*!

"*Ed io non so chi va e chi resta.*"
(Eugenio Montale)

Resumo

Progresso em Processamento de Linguagem Natural (PNL) tem sido ditado pela *regra de mais*: mais dados, mais poder de computação, mais complexidade, exemplificado pelos *Large Language Models*. Contudo, o treinamento (ou *fine-tuning*) de modelos grandes e densos para aplicações específicas geralmente requer quantidades significativas de recursos de computação. Uma maneira de lidar com esse problema é por meio da engenharia de dados (ED), em vez das perspectivas algorítmicas ou de *hardware*. Nesse contexto, nosso foco aqui é em uma técnica de ED pouco investigada, porém com enorme potencial no cenário atual – *Seleção de Instâncias* (SI). O objetivo do SI é *reduzir o tamanho do conjunto de treinamento removendo instâncias ruidosas ou redundantes enquanto mantém (ou melhora) a eficácia dos modelos treinados e reduz o custo do processo de treinamento.* Nesse sentido, a principal contribuição desta tese é dupla. Primeiramente, examinamos técnicas clássicas e recentes de SI e fornecemos uma comparação cientificamente sólida aplicadas a uma tarefa essencial de PNL - Classificação Automática de Texto (CAT). Os métodos SI têm sido normalmente aplicados a pequenos conjuntos de dados tabulares e não foram sistematicamente comparados na tarefa de CAT. Consideramos várias soluções CAT de última geração neurais e não neurais aplicadas a diversos conjuntos de dados. Respondemos a várias questões de pesquisa com base no *trade-off* do um tripé: eficácia, eficiência, redução. Nossas respostas revelam um enorme potencial para soluções de SI. Além disso, no caso de ajuste-fino dos métodos *transformers*, os métodos SI reduzem a quantidade de dados necessários, sem perder a eficácia e com ganhos consideráveis de tempo de treinamento. Considerando as questões reveladas pelas abordagens tradicionais de SI, a segunda principal contribuição é a proposta de duas soluções de SI. **E2SC**, um *framework* orientado a redundância de duas etapas destinada a grandes conjuntos de dados com foco particular em *transformers*. O E2SC estima a probabilidade de cada instância ser removida do conjunto de treinamento com base em classificadores fracos escaláveis, rápidos e calibrados. Nossa hipótese é que é possível estimar a eficácia de um classificador forte (transformer) com um mais fraco. No entanto, como mencionado, o E2SC concentra-se apenas na remoção de instâncias redundantes, deixando outros aspectos intocados, como o ruído, que podem ajudar a reduzir ainda mais o treinamento. Portanto, também propomos o **biO-IS**, um framework estendido construído sobre o anterior, com o objetivo de remover simultaneamente instâncias redundantes e ruidosas do treinamento. O biOIS estima a redundância com base no E2SC e captura o ruído com o suporte de uma nova etapa baseada na entropia. Também propomos um novo processo iterativo para estimar taxas de redução quase ótimas para ambas as etapas.

Nossa solução final é capaz de reduzir os conjuntos de treinamento em 41% em média (até 60%), mantendo a eficácia em todos os conjuntos de dados testados, com ganhos de aceleração de 1,67 em média (até 2,46x). Nenhuma outra linha de base foi capaz de escalar para conjuntos de dados com centenas de milhares de documentos e alcançar resultados com este nível de qualidade, considerando o compromisso entre redução, eficácia e aceleração do treinamento.

Palavras-chave: seleção de instâncias; classificação automática de texto.

Abstract

Progress in Natural Language Processing (NLP) has been dictated by the *rule of more*: more data, more computing power, more complexity, best exemplified by the Large Language Models. However, training (or fine-tuning) large dense models for specific applications usually requires significant amounts of computing resources. Our focus here is an under-investigated data engineering (DE) technique, with enormous potential in the current scenario – *Instance Selection* (IS). The IS goal is *to reduce the training set size by removing noisy or redundant instances while maintaining or improving the effectiveness (accuracy) of the trained models and reducing the training process cost.* In this sense, the main contribution of this Ph.D. dissertation is twofold. Firstly, we survey classical and recent IS techniques and provide a scientifically sound comparison of IS methods applied to an essential NLP task - Automatic Text Classification (ATC). IS methods have been normally applied to small tabular datasets and have not been systematically compared in ATC. We consider several neural and non-neural SOTA ATC solutions and many datasets. We answer several research questions based on tradeoffs induced by a tripod: effectiveness, efficiency, reduction. Our answers reveal an enormous unfulfilled potential for IS solutions. Furthermore, in the case of fine-tuning the transformer methods, the IS methods reduce the amount of data needed, without losing effectiveness and with considerable training-time gains. Considering the issues revealed by the traditional IS approaches, the second main contribution is the proposal of two IS solutions: **E2SC**, a novel redundancy-oriented two-step framework aimed at large datasets with a particular focus on transformers. E2SC estimates the probability of each instance being removed from the training set based on scalable, fast, and calibrated weak classifiers. We hypothesize that it is possible to estimate the effectiveness of a strong classifier (Transformer) with a weaker one. However, as mentioned, E2SC focuses solely on the removal of redundant instances, leaving other aspects, such as noise, that may help to further reduce training, untouched. Therefore, we also propose **biO-IS** an extended framework built upon our previous one aimed at simultaneously removing redundant and noisy instances from the training. biO-IS estimates redundancy based on E2SC and captures noise with the support of a new entropy-based step. We also propose a novel iterative process to estimate near-optimum reduction rates for both steps. Our final solution is able to reduce the training sets by 41% on average (up to 60%) while maintaining the effectiveness in **all** tested datasets, with speedup gains of 1.67 on average (up to 2.46x). No other baseline, was capable of scaling for datasets with hundreds of thousands of documents and achieving results with this level of quality, considering the tradeoff among training reduction, effectiveness, and speedup.

Keywords: instance selection; automatic text classification.

List of Figures

List of Tables

List of Acronyms

NLP	Natural Language Processing
IR	Information Retrieval
DE	Data Engineering
DL	Deep Learning
NN	Neural Network
IS	Instance Selection
FS	Feature Selection
ATC	Automatic Text Classification
AI	Artificial Intelligence
SOTA	State-of-the-Art
TF-IDF	Term Frequency — Inverse Data Frequency
MF	MetaFeatures
BERT	Bidirectional Encoder Representations for Transformers
XLNET	Generalized Autoregressive Pretraining for Language Understanding
RoBERTa	Robustly optimized BERT approach
GPT	Generative Pretrained Transformer
DistilBert	Distiled BERT
Albert	A Light BERT
BART	Bidirectional and Auto-Regressive Transformer
LSTM	Long Short Term Memory
GCN	Graph Convolutional Networks
KNN	K-Nearest Neighbors

SVM	Support Vector Machine
RF	Random Forest
NB	Naive Bayes
NC	Nearest Centroid
DT	Decision Trees
LR	Logistic Regression
XGBoost	Extreme Gradient Boosting
LGBM	Light Gradient-Boosting Machine
BS	Brier Score
CNN	Condensed Nearest Neighbor
ENN	Edited Nearest Neighbor
IB3	Instance-Based 3
ICF	Iterative Case Filtering
DROP3	Decremental Reduction Optimization Procedure 3
LSSm	Local Set-based Smoother
LSBo	Local Set Border Selector
LDIS	Local Density-based IS
CDIS	Central Density-based IS
XLDIS	eXtended Local Density-based IS
PSDSP	Prototype Selection based on Dense Spatial Partitions
EGDIS	Enhanced Global Density-based IS
CIS	Curious IS
E2SC-IS	Effective, Efficient, and Scalable Confidence-Based Instance Selection
biO-IS	Bi-Objective Instance Selection Framework
LLM	Large Language Model

Contents

Chapter 1

Introduction

1.1 Motivation

We have been experiencing an unprecedented increase in data availability, which has lead to enormous difficulties in organizing and retrieving such content in meaningful ways. Automatic Text Classification (**ATC**)[1] techniques are useful tools in this context by being able to map textual documents, such as web pages, emails, reviews, tweets, social media messages, etc., into a set of pre-defined categories of interest for a given application. ATC models have been demonstrated to be highly relevant given new difficult application scenarios such as the detection of fake news [112] and hate speech [91], relevance feedback [61], sentiment analysis [113], revision of product characteristics [88, 45, 49], inferring votes in elections [59], assessing satisfaction with government agencies [38], among many others. Being a supervised task, ATC has benefited from applications that constantly produce high volumes of (labeled) data (e.g., large-scale social networks, such as Twitter), in which users can manually classify messages, advertisements, and products, producing a large volume of annotations [57]. The costs of obtaining large amounts of labeled data can also be ameliorated by approaches such as crowd [125] and soft labeling [116].

Currently, Transformer-based architectures (including 1st and 2nd generation Transformers such as RoBERTa [85] and BART [78] as well as current Large Language Models such as GPT4 [13] and LLama3 [131]) stand out as the state-of-the-art (SOTA) in ATC, achieving remarkable results across various tasks [89, 90, 3, 95]. In more details, these deep learning approaches can be divided into two steps: (i) pre-training; and (ii) domain transfer. The pre-training step involves learning the model weights employing an unsupervised task (e.g., Next Sentence Prediction [44]). The fine-tuning step is supervised, applied to a domain-specific labeled dataset and allows for further model optimization. In order to achieve such performance, these models rely on huge training sets and complex architectures with millions of parameters [141]. While these models can exhibit some degree of effectiveness when used in a zero-shot manner, their fine-tuning for specific domains or tasks is crucial to ensure increased performance [41].

[1] In this dissertation, we focus on both binary and multi-class single-label classification by addressing two types of tasks: i) topic categorization and ii) sentiment analysis (polarity detection).

Indeed, according to Andrew Ng [103], there are two main reasons for the successful results. The first one is the amount of data used to pre-train these models – the GPT-3 model [13], for instance, was pre-trained on 45TB of textual data. The second reason is the possibility of reusing and adapting the general pre-trained model in multiple tasks by just fine-tuning the model's last layers for the specific task, which is considerably faster than training from scratch for each task.

Among the challenges related to these approaches, we can mention: (i) the need for **large**[2] amounts of annotated (manually classified) data to perform effective learning; and (ii) issues related to the scalability of the solutions (even in the fine-tuning stage) in the face of collections with millions, sometimes billions, of documents.

Regarding the first challenge, several applications that constantly produce data, such as social networks (e.g., Facebook and Twitter), with a high number of users, have tools that help the end user to manually classify messages, advertisements, products, etc. Moreover, crowdsourcing [125] and soft labeling [116] annotation (labeling) methodologies are also possible solutions for acquiring large amounts of labeled data with reduced costs. These types of labeling strategies have gained notoriety [57] for allowing a large volume of annotations without the help of experts. ChatGPT[3] – the novel OpenAI artificial intelligence (AI)-empowered virtual assistant – is an example of a system that utilized crowdsourcing for annotation. However, the main disadvantage of these strategies is they are very prone to noise – in the context of classification tasks, represented mainly as instances of the training set assigned to the wrong classes. Indeed, the authors in [94] show that in review domains, almost a quarter of the instances (23%) are considered difficult to classify, even for humans. In addition, users (regulars or experts) make mistakes in classifying these difficult instances between 56%-64% of the time. In the case of the ChatGPT tool, the noise was possibly aggravated as the Kenyan workers responsible for the annotation were notoriously underpaid, earning less than $2 per hour [29]. Thus, noise data instances can potentially constitute a large portion of the available data in these scenarios.

Regarding the second challenge, despite faster, fine-tuning is still a costly process that demands expensive computational resources in terms of computational power and memory demands. For instance, as we shall see in our experiments, the fine-tuning process on the MEDLINE dataset, used in our experiments only for one transformer (XLNET), takes approximately 80 hours of uninterrupted processing using specialized GPU hardware. Indeed, from a practical point of view, there are several scenarios in which adopting fine-tuned deep learning approaches can be very difficult (if not impractical) despite potential effectiveness gains. For instance, consider a textual classifier applied in a scenario that requires continuous (constant) re-training (e.g., fraud detection [86], product tagging [9],

[2] In this work, we adopted the definition given in [150], where the authors defined large-scale datasets for the automatic text classification task as datasets with the number of samples ranging from hundreds of thousands to several millions of documents (more than 100,000 documents).

[3] https://chat.openai.com/

and recommendation [25]). Due to the continuous changes in the data stream source, these models need constant re-training to reflect modifications in the interest domain. Constantly re-training (fine-tuning) the model, as mentioned, can be very costly – computationally and financially. The practical solution is usually increasing the time between consecutive model training (a.k.a. training window), delaying the learning of the temporal changes in the input data, which, in turn, can affect the effectiveness of the task [101].

Another practical scenario is the challenge posed by using deep learning models in the context of companies and research groups with financial budget constraints. In both contexts, the application and experimentation of these models are limited to the available resources. Moreover, there is often the need to run thousands of experiments to propose scientifically-sound or practical (commercial) advances regarding the SOTA. For instance, for this Ph.D. dissertation, we run **four thousand** experiments using SOTA Transformers corresponding to about **5,600** hours (233 days) of experiments. Any reductions could bring benefits from several perspectives (financial, energy, etc.).

Another issue related to the cost of training or fine-tuning a deep learning model is carbon emission. The amount of energy and time required to perform the parameters adjustment to optimize the models' effectiveness can vary depending on several factors, including: (i) the size and complexity of the model; (ii) the use of specialized hardware, such as GPU and TPU (energy demanding); and (iii) the amount of data. As most of the world's electricity is generated using fossil fuels [39], the process above can be considered directly responsible for releasing carbon dioxide into the environment. In addition, the study conducted by Patterson et al. [106] highlights that the pre-training phase of GPT-3 consumed 1,287 MW/h. This energy consumption resulted directly in the emission of at least 552 tons of CO2e. To put this in perspective, this is equivalent to the carbon footprint generated by running a car for 1.3 million miles, according to the ML CO2 Impact calculator[4]. Furthermore, the authors estimated that the ChatGPT's daily carbon footprint to maintaining the core model is approximately 23 kgCO2e, making it crucial to address the environmental impact of Transformer models.

Given these scenarios of ever-expanding volumes of data with constant re-training requirements, budget constraints, and high-demanding energy models, it is desirable to develop new effective, effective, and scalable strategies to handle those issues properly. Two (costly) alternatives are developing new deep learning algorithms or more efficient hardware. Another way to ameliorate these problems is through data engineering [31], which may be achieved by (not mutually exclusive): (i) Model Compression (or Pruning) techniques [82, 102] applied to reduce the complexity of the DL models; (ii) Data Preprocessing techniques [123], aimed at improving the quality of the input training data for the ATC models. The latter focuses on improving performance while reducing training time and computational costs. In this dissertation, we focus on this second alternative.

[4] https://mlco2.github.io/impact/

In [31], we proposed exploiting a set of preprocessing techniques in a data transformation pipeline for building cost-effective models. That solution achieved improved effectiveness (e.g., models with higher accuracy) at a much lower cost (e.g., shorter time for ATC model construction). One of the main contributions of [31] was the explicit incorporation into the pipeline of an **Instance Selection** stage[5], a promising set of techniques and growing research area that helps to deal with many of the aforementioned issues.

In contrast to traditional Feature Selection approaches, in which the main objective is to select the most informative terms (words), **Instance Selection** methods are focused on selecting the most representative instances (documents) for the training set [55]. The intuition behind this kind of algorithm is to remove potentially noisy or redundant instances from the original training set and improve performance in terms of total time training time while keeping or even improving effectiveness.

More specifically, IS methods have three main goals: (i) to reduce the number of instances by selecting the most representative ones; (ii) to maintain (or even improve) effectiveness by removing noise[6] and redundancy; and (iii) to reduce the total time for applying an end-to-end model (which includes from traditional preprocessing steps to the model training step). By selecting the most representative instances, IS methods can also potentially remove noise from erroneous annotations. According to these objectives, IS methods must respect three fundamental constraints – tripod-constraints set illustrated in Figure 1.1 – consisting of *reducing the amount of training without loss of effectiveness and with efficiency gains*. IS methods seek to optimize these three constraints simultaneously. It is important to note that these are conflicting constraints and that some of them may not be achieved simultaneously in many situations. For example, a quick selection method could randomly select class instances according to the classes´ distribution in the training set. Despite being fast in the selection step, this simple approach would probably suffer regarding the effectiveness of the generated models.

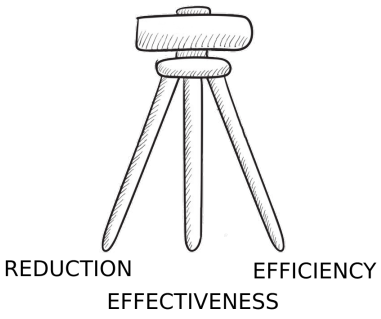


Figure 1.1: Tripod-constraints set: Tradeoff among reduction, efficiency, and effectiveness

[5]Instance Selection (IS) is also known in the literature as Selective Sampling, Prototype Selection, and Instance-Based. For didactic reasons, we will refer to the subject hereafter only by Instance Selection.

[6]We defined noise as instances incorrectly labeled by humans in the dataset [94] as well as (possible) outliers that do not contribute (or even get in the way) to model learning (tuning).

Despite their immense potential, there are just a few studies about IS approaches in the context of ATC [6, 132], especially in the deep learning scenario. Traditional methods that have been proposed in the literature for scenarios other than ATC include: *Condensed Nearest Neighbor* (**CNN**)[62], *Edited Nearest Neighbor* (**ENN**)[143] and *Decremental Reduction Optimization Procedure* (**Drop3**)[142]. More recent approaches include: *Local Set-based Smoother* (**LSSm**)[79], *Local Set Border Selector* (**LSBo**) [79] and *Prototype Selection based on Dense Spatial Partitions* (**PSDSP**)[20]. Indeed, most of the IS methods have been proposed and studied only on small tabular datasets (e.g., toy examples from UCI[7]), and the selected instances were applied as input only to weak classifiers, such as KNN (Chapter 2). In contrast, the datasets in text classification are unstructured, larger, and more complex, with high dimensionality and potentially high skewness. As deep learning transformer approaches have a high cost in terms of computational resources, mainly when dealing with large training data, we believe they constitute an ideal scenario for applying IS techniques.

Accordingly, in this work (Chapter 3), we extensively study the behavior of these IS techniques in ATC tasks, initially delimiting the scope to multiclass and single-label classification tasks of textual documents. More specifically, we propose to study the use of IS strategies in two types of tasks: (i) classification into semantic topics; and (ii) sentiment classification (polarity detection). The studied datasets[8] are widely used in the literature (ATC benchmarks) and cover various sources and domains, including web pages, questions and answers (Q&A), news, comments, reviews, social networks, etc.

We also perform an original investigation regarding the impact of IS methods on state-of-the-art deep learning Transformer approaches (e.g., BERT, RoBERTa, BART, and GPT). Neural network algorithms, especially Transformers, currently achieve the best results in several benchmarks used by the scientific community. To achieve such state-of-the-art performance, they usually rely on large amounts of data in the fine-tuning training stage of a neural network model [103, 76, 46] – which is diametrically opposed to the philosophy of IS approaches. Thus, investigating whether IS approaches are applicable and useful for these large deep-learning models is an interesting research goal in itself.

As far as we know, a study of the magnitude and rigor of ours, covering a large set of IS methods applied to the most recent neural and non-neural ATC solutions, has not been reported in the literature yet. Indeed, most of the comparative IS results presented here have not been reported elsewhere. Given that several applications (e.g., news portals, search engines, market research, among others) exploit ATC models as a basis, our results may have a direct practical impact on the use of complex ATC models in real applications.

[7] https://archive.ics.uci.edu/ml/datasets.php

[8] All used datasets and our code are publicly available on https://github.com/waashk/instanceselection

Besides, considering the aforementioned application potential as well as the obtained results in Chapter 3, we propose in Chapter 4 and Chapter 5, two novel IS methods able to handle the aforementioned ever-expanding volumes of data with constant re-training requirements and budget constraints domains, with a particular focus on Transformer-Based architectures. Further details are in the next Sections.

1.2 Objectives

The goal of this Ph.D. dissertation is threefold. Our first objective is to conduct a comprehensive literature review of both traditional and state-of-the-art methods in the IS field to gain a thorough understanding of various IS approaches and the advancements made in recent years. In addition to the literature review, we aim to extend the existing 10-year-old taxonomy [56] of IS strategies, delivering a thorough up-to-date categorization of IS strategies along with their respective strengths and limitations.

Our second objective is to experimentally compare the traditional and SOTA IS methods (Chapter 3), given the significant evolution in this field over the past years and the absence of a recent experimental comparison – the most recent one has been published almost a decade ago. Our objective is to study the impact of using IS methods in an important NLP area – ATC – in which IS methods have a large potential due to the current computational costs of the SOTA methods in this field and the increasing volume of textual data to be classified. In particular, regarding the previous objective, we aim to study and analyze the trade-offs among training set reduction, efficiency, and effectiveness (*tripod-constraints set* – Fig. 1.1) of traditional and SOTA IS methods for Topic Classification and Sentiment analysis tasks.

Finally, our third objective is to address issues and problems revealed by the results of our previous comprehensive experimental comparison. More specifically, scalability, redundancy and noise removal. In this sense, we propose two novel IS solutions. Firstly, we propose a redundancy-oriented IS framework aimed at large datasets with a particular focus on Transformer-Based architectures that is able to reduce the training sets while maintaining the same levels of effectiveness regardless of the context, with speedup improvements, scaling even for datasets with hundreds of thousands of documents (something that the traditional IS methods cannot do). Next, we propose an extended IS framework built upon our first one aimed at simultaneously removing redundant and noisy instances from the training set. This extended proposal involves a comprehensive analysis of additional metrics and techniques for noise removal aiming to improve the overall performance of our IS framework. This extension was motivated by the fact that the first proposed framework was not designed to effectively handle nor remove noise, which is a potential issue given the aforementioned scenarios of crowd-sourced and user-generated labeling.

1.3 Hypothesis, Research Questions and Findings

The main hypothesis (H1) of this Ph.D. dissertation is:

H1: It is possible to simultaneously reduce data, maintain model quality, and improve time for fine-tuning ATC models through IS methods.

In order to confirm this hypothesis, we propose **three** research questions for our Ph.D. dissertation. In sum, **RQ1** aims at evaluating the traditional IS approaches presented in Chapter 2 in the context of the ATC task concerning the posed IS methods tripod constraints: reduction, effectiveness, and efficiency. Next, considering the literature gap, **RQ2** aims at demonstrating the feasibility of proposing a novel IS framework and by showing how it can accommodate different requirements posed by distinct scenarios, mainly those associated with big data. Finally, **RQ3** aims to investigate the capability of each IS method to handle and effectively remove noisy instances. The answer to this question also motivated us to demonstrate the feasibility of proposing a novel extended IS framework capable of removing simultaneously redundant and noisy instances from the training set. Next, we present each of the RQs considered in this Ph.D. dissertation in depth as well as an overview of the main findings of each one of them.

RQ1. ***What is the impact of applying traditional IS methods in the ATC context regarding the posed constraints?*** RQ1 aims to evaluate the traditional IS approaches in the context of the ATC task, focusing on the tripod constraints of the posed IS methods: reduction, effectiveness, and efficiency.

In order to conduct a thorough evaluation of each constraint, we have divided this RQ into three incremental sub-questions:

RQ1.1. ***Are there traditional IS methods capable of reducing the training set while keeping classifier effectiveness for each investigated scenario?*** The objective of RQ1.1. is to investigate the tradeoff between the first two constraints of the "tripod": effectiveness and reduction.

As result, we found the IS methods can, in some cases, reduce the training set by up to 90% while maintaining effectiveness. The studied IS approaches achieved average reductions between 15.6% (LSSm) to 91.1% (XLDIS). On the other hand, despite the potential for noise removal motivation, selection methods were not able to improve the effectiveness of the text classification models in none of the tested textual datasets. We suggest further studies in the future to investigate this last issue.

RQ1.2. ***What is the impact of applying IS strategies on the text classification models' total construction time?*** We propose to evaluate the impact of IS (studied in RQ1.1. in terms of effectiveness and training set reduction) regarding potential speedups on the time to complete the full pipeline process, which corresponds to the sum of times of preprocessing steps (including IS step) and ML training model.

We found that three traditional IS methods **(LSSm, CNN, LSBo)** were able to reduce the total text classification models' construction time while keeping the effectiveness in 12 (out of 19) considered datasets – with speedups between 1.04x (CNN - Books) and 5.69x (LSBo - Reuters90). In the other datasets, we observed that the introduction of IS approaches caused an overhead in terms of the total time to generate the model (running the IS methods + model construction), making the whole process more costly from a computational cost perspective. Overall, considering the three tripod constraints altogether and all datasets, the best traditional IS method was **CNN**.

RQ1.3. ***How do IS approaches behave when applied to neural classification methods regarding the tripod constraints?*** This question investigates and challenges the widespread notion that fine-tuning DL classifiers require a large amount of labeled data. Our goal is to investigate whether IS methods can work with NN solutions in face of the current anecdotal and empirical evidence to the contrary.

Our experiments confirm the importance of fine-tuning neural-based models in the context of ATC to obtain good effectiveness, which is consonant with previous work [34]. Since a large part of the execution time of deep learning models is associated with the fine-tuning phase, the application of IS approaches is promising. For instance, according to our analyses, the best selection method (CNN) can reduce the training set in 11 datasets, producing speed-ups of model construction time between 1.04x and 3.24x while maintaining effectiveness. Thus, we can show that deep learning networks do not always need massive training data for fine-tuning and that a carefully selected training set may be enough to produce effective models. Indeed, similar effectiveness results were obtained with reductions ranging up to 72%. This last result may have important implications for the practical application of large neural network methods, especially Transformer architectures.

Therefore, answering **RQ1**, our evaluation of the tripod constraints (reduction - efficiency - effectiveness) of several traditional IS methods demonstrates that in the majority of the cases - 12 out 19 – specific IS methods - namely LSSm, CNN, LSBo – can reduce the size of the training set without effectiveness losses, leading to efficiency improvements. Specifically, in the case of fine-tuning the transformer methods, the IS methods reduce the amount of data needed, without losing effectiveness and with considerable training-time gains. However, there is also a significant number of cases in which the requirements of the tripod cannot be fully satisfied by any traditional method. Moreover, traditional IS strategies demonstrated not scaling for the big data scenario (e.g., datasets with more

than 100K instances). Therefore, these findings neither totally support nor completely refute our posed hypothesis. Our experiments indicate an affirmative answer for RQ1 – there are traditional IS methods are capable of simultaneously reducing data, maintaining model quality, and improving time for fine-tuning models. These results highlight that further investigation of IS methods applied to the ATC context is needed, particularly concerning recent transformer architectures. In any case, our study concerning RQ1 reveals that there is a lot of room for developing more efficient, effective, and scalable IS methods for the big data scenario in general. This answer leads to the following RQ:

RQ2. ***Can a novel instance selection method focused on redundancy removal overcome the limitations of existing IS methods to achieve the tripod restrictions in the ATC scenario?*** This RQ aims to demonstrate the feasibility of proposing a novel IS framework and by showing how it can accommodate different requirements posed by distinct scenarios, mainly those associated with big data.

In order to provide a more efficient, effective, and scalable IS method – addressing the posed research question RQ2 – we propose the **E2SC** framework, our first redundancy-oriented IS solution. We compare E2SC proposal with **six** robust state-of-the-art instance selection baseline methods considering as input of the best of **seven** deep learning text classification methods in a large benchmark with **19** datasets. Our experimental evaluation show that **E2SC** managed to significantly reduce the training sets (by **27%** on average; varying between 10% and 60% of reduction) while maintaining the same levels of effectiveness in **18** (out of 19) considered datasets. Also, we found that **E2SC** was able to reduce the total text classification models' construction time while keeping the effectiveness in all (19) considered datasets – with speedups of **1.25** on average, varying between 1.02x (Books) and 2.04x (yelp_reviews). Overall, considering the three tripod constraints altogether and all datasets, the best IS method so far was our first proposed framework. Finally, to demonstrate the flexibility of our framework to cope with large datasets, we propose two modifications. Our enhanced solution managed to increase the reduction rate of the training sets (to **29%** on average) while maintaining the same levels of effectiveness in **all** datasets, with speedups of **1.37** on average. In addition, the framework scaled to large datasets, reducing them by up to 40% while statistically maintaining the same effectiveness with speedups of **1.70x**.

Despite being innovative and achieving significant results in terms of effectiveness, efficiency, and reduction, the E2SC framework focused only on **redundancy**, leaving some other aspects that may help to further reduce training untouched. One such aspect is **noise**, here defined as instances incorrectly labeled by humans in the dataset [94] as well as (possible) outliers that do not contribute (or even get in the way) to model learning (tuning). Indeed, according to [94], users, whether regular individuals or experts, make a reasonable amount of mistakes while labeling complex instances – between 56% and 64%

of the time. Other few noisy instances (a.k.a., outliers) may be correctly labeled, but they differ so significantly from other instances from the same class that they are either useless for the sake of learning (tuning) or may even be detrimental to the process.

Noisy instances can potentially constitute a significant portion of available data in these contexts. Noisy (training) instances may not only degrade the model's effectiveness by incorporating misleading patterns in the model but may also be detrimental to performance as they need to be processed to extract and incorporate these patterns into the model. If the amount of noise is significant, there will certainly be negative impacts on effectiveness and efficiency. However, in a simulated scenario designed to evaluate the capability of the IS baseline methods and our previous solution to remove noise, none of the IS solutions satisfactorily performed the task. This answer leads to the following RQ:

RQ3. ***Is it possible to extend the previous proposal to not only remove redundancy but also remove noise, enhancing the level of quality considering all tripod criteria?*** The objective of this RQ is to demonstrate the feasibility of proposing a novel extended IS framework capable of remove simultaneously redundant and noisy instances from the training set.

In order to remove simultaneously **redundant** and **noisy** instances from the training – addressing the posed research question RQ3 — we propose the extended **biO-IS** framework, our ultimate IS solution. We compare **biO-IS** with **seven** robust state-of-the-art instance selection baseline methods, including our first proposal, E2SC, in the text classification domain considering the same benchmark covering **22** datasets. Our experimental evaluation reveals that, in a simulated scenario designed to evaluate the capability of the IS baseline methods and our previous solution to remove noise, none of the IS solutions were capable of satisfactorily performing the task. On the other hand, **biO-IS** managed to remove up to 66.6% of the manually inserted noise. Moreover, **biO-IS** managed to significantly reduce the training sets (by **40.1%** on average; varying between 29% and 60% of reduction) while maintaining the same levels of effectiveness in **all** of the considered datasets. Also, **biO-IS** managed to consistently provide speed-ups of **1.67x** on average (maximum of **2.46x**). No baseline, not even our previous SOTA solution, was capable of achieving results with this level of quality, considering all tripod criteria. Indeed, the only other method capable of maintaining the effectiveness on all datasets was E2SC; **biO-IS** improves over E2SC in 41% regarding reduction rate and from 1.42 to 1.67 (on average) regarding speedup, achieving the SOTA in the Instance Selection field.

1.4 Contributions

Our work in the Instance Selection field has been validated and published in the main Information Retrieval (**IR**) and Natural Language Processing (**NLP**) conferences and journals in the last four years only, including **two** published papers in the *Information Processing and Management* **(IP&M)** *(h-index: 123, Impact Factor: 8.6, A1)*, a worldwide leading journal in *Information Retrieval* [34, 31], covering respectively: (1) a comprehensive comparative study of the cost-effectiveness of neural and non-neural approaches and representations for ATC; (2) three new steps (MetaFeatures, Sparsification, and Instance Selection) into the traditional pre-processing phase of pipelines for text classification as well as a thorough and rigorous evaluation of the trade-offs between cost and effectiveness associated with the introduction of these new steps. These first two articles helped to define the scope of the Ph.D. dissertation.

There is also a publication in *ACM Computing Surveys* **(CSUR)** *(h-index: 213, Impact Factor: 16.6, A1)* [36], a worldwide leading journal in *Computer Science Theory and Methods*, covering comprehensive literature review of both traditional and SOTA methods in the IS field as well as an extensive experimental comparison between these methods; the *International Conference on Research and Development in Information Retrieval* **(SIGIR)** [33], covering our first proposal of a novel redundancy-oriented IS framework aimed at large datasets with a particular focus on transformers. Finally, this dissertation also resulted in a submitted paper (under review) to the *ACM Transactions on Information Systems* (**TOIS**) *(h-index: 95, Impact Factor: 5.6, A1)*, which includes the proposal of an extended noise-oriented and redundancy-aware IS framework for ATC.

Furthermore, this dissertation led to several international collaborations. Specifically, a period was spent abroad under the supervision of Professor Fabrizio Sebastiani, focusing on the investigation of IS strategies through Feature Selection. The collaboration with Prof. Fabrizio enabled connections with two other globally respected research groups headed by Prof. Nicola Ferro from the University of Padua and Prof. Davide Bacciu from the University of Pisa. Through the collaboration with Prof. Ferro, we had a full paper accepted at the *International Conference on the Theory of Information Retrieval* (**ICTIR**) [105], introducing an innovative Quantum Annealing (QA) IS approach. As far as we know, our solution was the first to apply QA to the IS problem and it offers a new *Quadratic Unconstrained Binary Optimization* formulation. Our collaboration with Prof. Bacciu has also led to the establishment of a research project, currently in its early stages.

This dissertation also resulted in several undergraduate students advising. We highlight the work present in [51], accepted in the Scientific Initiation Paper Competition (CTIC) in the annual Congress of the Brazilian Computing Society (CSBC) covering the proposal of instance selection-inspired **Undersampling** strategies for bias reduction in the context of transformer-based text classification – work co-advised by the propo-

nent. This work also resulted in a submitted paper (under review) to the *Conference on Empirical Methods in Natural Language Processing* (**EMNLP**).

In addition to these works directly obtained from this Ph.D. Dissertation, in this doctoral period, we also contributed to another **(IP&M)** [41] paper studying how BERT-based contextual representations effectively improve NLP tasks, particularly ATC, which is directly related to highly separable characteristics that allow the simplest classifiers to achieve high effectiveness. This work also contributed to other papers in important conferences and journals, such as **Neurocomputing** [54] *(h-index: 196, IF: 5.5)*, **JMIR** [147] *(h-index: 197, IF: 7.4)*, **CIKM** [97], **ACL** [137], **WSDM** [136], **SBBD** [119], **JIS** [135], **WebMedia** [69, 134, 22, 52, 124], and **Value in Health** [146].

Finally, for the sake of reproducibility, we make the documented code of all compared methods (IS and classifiers) as well as the preprocessed and raw datasets, including the division in folds, available to the community for replication and further comparisons. We consider that making the code used in our experimental protocol, including the methods we had to implement ourselves, along with the datasets and the appropriate documentation is very useful for reproducibility and comparison of future IS and ATC methods.

1.5 Roadmap

The remainder of this Ph.D. dissertation is organized as follows.

Chapter 2 This chapter covers a literature review of the most traditional and/or current IS methods and provides a new extended IS taxonomy. Also, we describe the relevant works from the IS literature based on the retrieved approaches from the literature review, presenting their methodology, strengths, and weaknesses.

Chapter 3 This chapter provides a thorough and comprehensive cost-effectiveness survey by applying the IS methods retrieved in the previous chapter in the context of automatic text classification.

Chapter 4 This chapter introduces **E2SC** – **e**ffective, **e**fficient, and **s**calable **c**onfidence-Based **i**nstance **s**election – a novel two-step framework aimed at large datasets with a particular focus on transformer-based architectures, our first IS proposal.

Chapter 5 In this chapter, we introduce **biO-IS** – an extended **bi**-**o**bjective **i**nstance **s**election framework built upon our previous one aimed at simultaneously removing redundant and noisy instances from the training, our second IS method proposal.

Chapter 6 Ultimately, in this chapter, we conclude the Ph.D. dissertation, summarizing our main findings and proposing some directions for further investigation.

Chapter 2

Systematic Literature Review of Instance Selection Methods

In this chapter, we present a critical analysis (*a.k.a.,* rapid (systematic-based) literature review[1]) of the most traditional and/or recent (state-of-the-art) proposals in the Instance Selection (IS) area. The objective of this review is to comprehensively assess the most relevant works related to IS strategies applied in different scenarios. In particular, we focus on experimentally-oriented studies, that is, studies that have strong experimental and empirical components to support their findings.

To achieve our objective, we collected a set of 100 publications[2] that included the most cited articles related to IS. We assume that highly cited articles are potentially influential as they have received much attention. From those, we selected the most popular methods to include in our experimental assessment of IS methods applied to ATC. We also selected a set of recently proposed methods (considered state-of-the-art) to complete our experimental comparison. At the end of this literature review process, which is further detailed next, we end up with a mix of the traditional and state-of-the-art set of methods, comprising 13 IS strategies to be evaluated in the next Chapter in the ATC scenario.

A simplified version of the rapid review procedure is shown in Figure 2.1. First, we collected articles returned by a set of four queries submitted to Google Scholar. Second, we used each article's unique URL to remove duplicates (deduplication phase). This procedure resulted in 1,740 unique articles [2]. Third, we ranked the remaining articles by the number of citations. Fourth, we classified the articles, in ranked order, according to the desired criteria of (i) being related to IS and (ii) conducting experimental comparisons among methods. Fifth, we filtered out articles that did not match the criteria, up to the point that we achieved 100 relevant articles [3]. From these articles, we selected the most popular and most recent methods to be compared in our experimental assessment. The remainder of this chapter provides detailed information for each of the phases, depicted in Figure 2.1 and the process of selecting the IS methods.

[1] From now on, we will use the terms 'literature review', 'rapid review' and 'critical analysis' interchangeably.

[2] The list of articles can be found at https://shorturl.at/zCLW7

[3] Up to this point, we had inspection-ed 702 articles in ranked order.

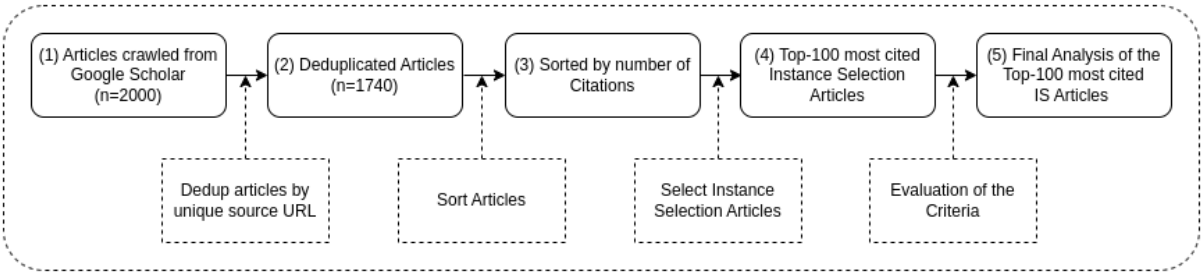


Figure 2.1: Workflow of study selection and analysis of literature (rapid) review.

2.1 Collecting and Selecting Relevant Articles

We used the Google Scholar[4] search engine to submit four queries to generate our initial set of articles. Google Scholar was our choice for two main reasons: (i) high coverage - it includes digital libraries from prestigious publishers such as ACM, IEEE, and Elsevier, as well as preprint repositories such as Arxiv; and (ii) it contains information about the number of citations for each research paper, an important criterion for our selection of method. To be more precise, we issue the following four distinct queries[5] : "*Instance Selection*", "*Selective Sampling*", "*Prototype Selection*", and "*Instance Based*".

These queries were issued to the search engine without any venue or year filter. We chose this setting to maximize the coverage of our queries. We gathered the first 500 papers (sorted by citations) for each submitted query, totaling 2000 papers using this approach. We deduplicated this list of papers by using the unique papers' URLs. After filtering out duplicates, we ended up with 1,740 unique papers ordered by the number of citations, as mentioned before.

We then manually classified the papers, in the ranked order, into two categories:

- **RELEVANT:** consisting of papers whose main objective is to propose and/or evaluate an IS method (experimentally-oriented study).
- **IRRELEVANT:** Non-experimental or not related to the IS subject (i.e., papers that do not fit into the aforementioned category).

We continue the process up to the point in which we selected the 100^{th} RELEVANT paper, which corresponds to position 702^{nd} of the ranked list by citations.

[4] https://scholar.google.com/

[5] Despite having different objectives, the areas of IS and undersampling are related, as both deal with techniques that aim to select a subset of representative data. Therefore, another potential query would be "undersampling". Indeed, considering this query, in another work [51], we systematically mapped the literature on undersampling methods, identifying and implementing 14 methods among the most popular and used, evaluating them in conjunction with Transformers classifiers from four perspectives: (1) classification effectiveness; (2) efficiency (time); (3) generalizability (bias); and (4) scalability. However, as our focus in this dissertation is IS, we chose not to include these results in this dissertation. The results of this parallel work can be found in [51].

It is worth emphasizing that all 100 papers are on the main subject of interest and are based on experimental work, as opposed to purely theoretical work. Among the selected 100 articles, the least cited has 21 citations, and the most cited has 7,158 citations.

2.2 Criteria for Selecting the IS methods

We examined the 100 selected publications, paying close attention to the IS methods used in each paper's experimental section. Two volunteers analyzed each paper to double-check the presence of each IS method. Three computer science researchers with contributions to IR and machine learning made up the volunteer group.

Summary of the Results We found five highly used methods in experimental comparisons:

- More than half (56%) of the analyzed top-100 relevant papers use the Decremental Reduction Optimization Procedure 3 (DROP3) method in their experiments.
- The Condensed Nearest Neighbor (CNN) is present in 44% of the analyzed papers.
- Edited Nearest Neighbor (ENN) and Instance-Based 3 (IB3) are used as a baseline in 34% and 30% of the selected papers, respectively.
- 26% of the analyzed papers use Iterative Case Filtering (ICF) as a baseline.

Accordingly, in our experimentation evaluation of IS methods for ATC, we consider DROP3, CNN, ENN, IB3 and ICF.

2.2.1 Increasing the Coverage and Representativeness of IS methods

As mentioned, our goal with this literature review is to select the most relevant IS strategies to evaluate them thoroughly in the ATC context. However, several of the most popular methods selected for our study are quite old. ICF, for instance, was proposed 20 years ago (2002).

In the meantime, the advantages of IS methods have been increasingly perceived by the research and practitioner communities, especially given the rise of expensive neural methods and the availability of large datasets. This has led to the proposals of new

methods in the last few years (5-7 years), which, by being newer, are not as popular or cited as the ones we chose in the previous step.

Thus, to increase the coverage and representativeness of the IS methods, and to reflect the field's evolution in our experimental evaluation, we selected several recent proposals in the IS area, some of which can be considered state-of-the-art in the field, according to the procedure explained below.

Method	**Year**
LSSm: Local Set-based Smoother	2015
LSBo: Local Set Border Selector	2015
LDIS: Local Density-based IS	2016
CDIS: Central Density-based IS	2016
XLDIS: eXtended Local Density-based IS	2017
PSDSP: Prototype Selection based on Dense Spatial Partitions	2018
EGDIS: Enhanced Global Density-based IS	2020
CIS: Curious IS	2022

Table 2.1: Instance Selection Recent Proposals

To avoid re-evaluating irrelevant papers, we removed the 602 papers considered irrelevant by the previous analysis from our list of 1702 deduplicated articles (Phase 2). Thus, we were left with 1102 papers: 1002 non-analyzed papers, and the 100 most cited articles related to IS.

Since our goal now is to reflect the field's evolution, focusing on the newest and state-of-the-art methods, we applied a filtering procedure in which we removed, from the set of 1102 papers, those older than ten years. We assumed that relevant methods to be used in our experiments, which have been proposed by papers published more than ten years ago, had been most probably selected in our previous step. This filtering procedure resulted in 636 papers to be analyzed.

Next, we (re)sorted the list of 636 papers now by their Google Scholar rank (GSRank[115, 7]) scores. Given a query, the GSRank score corresponds to a paper's position in the ranked list returned by Scholar. As we have four queries, for papers that appeared in more than one list, we used the highest position in any of them.

We then proceeded to manually classify the papers, by reading and analyzing each paper individually, in the ranked order of the GSRank score. We considered as RELEVANT, papers: (i) whose main objective is to propose and/or evaluate an IS method (experimentally-oriented study); (ii) in which the proposed IS method is compared with at least 3 of those previously cited (most popular methods). The second rule aims at selecting proposals that have been adequately evaluated in their context. We consider that a proposal was well evaluated if it was compared to the most traditional methods in the IS area. This final filtering resulted in eight recent proposals (considered) state-of-the-art. These methods along with the year they were proposed are presented in Table 2.1.

2.3 An Extended Taxonomy of IS Strategies

We propose a new taxonomy of IS strategies by extending a 10-year-old taxonomy proposed in [56]. This previous IS methods taxonomy was proposed in a different context, considering different types of datasets (small tabular ones) and different learning methods. For instance, deep learning neural network methods were not considered in the envisioned scenarios by the time that taxonomy was proposed.

As mentioned, the field has significantly evolved since the proposal of the original taxonomy. These advances are reflected in our new extended taxonomy with three new categories – represented in green in Figure 2.2 – and eight new, recently proposed methods – written with black color letters in Figure 2.2. The new categories refer to approaches based on density, spatial hyperplanes, and clustering-based approaches proposed since 2015s. Following we present a brief discussion for each category – previously existent and new ones.

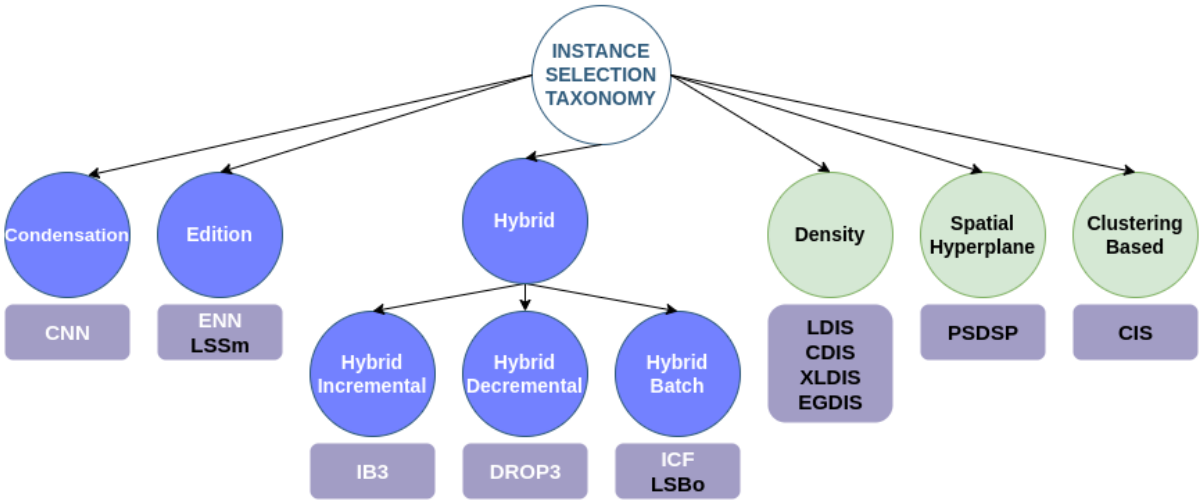


Figure 2.2: Instance Selection Taxonomy

Condensation algorithms perform noise removal by creating data subsets, which are used later to reduce the number of instances. The Condensed Nearest Neighbor (CNN) [62] is one of the most well-known approaches within this category. **Edition** algorithms perform removing noisy instances by employing filters. The most traditional method in this category is Edited Nearest Neighbor (ENN) [143]. The more recently proposed Local Set-based Smoother (LSSm) [79] (2015) also falls in this category. **Hybrid** algorithms combine the condensation and editing paradigms. Hybrid algorithms can be subdivided into three approaches: Incremental, Decremental, and Batch. The **Incremental** approach starts with an empty subset S and, for each instance s in the training set, s is inserted in the final subset S if meets certain requirements. **Decremental** algorithms perform the opposite task of incremental ones. The subset S starts containing every

training instance, and an instance s is removed from S if some requirement is fulfilled. The **Batch** approaches also start with a subset S containing the whole set of training instances. However, differently from **Decremental**, it proceeds by deciding whether each instance satisfies the removal criteria before eliminating any instances. In this way, Batch approaches first mark each instance to be removed or not. Then, all those instances that do meet the requirements are immediately eliminated. IB3 [2], DROP3 [142], ICF [12] and more recent LSBo [79] (2015) are the most referenced approaches within this category.

We extend this taxonomy [56] with three new categories. The first one is composed of **Density** algorithms that attempt to select instances by measuring the hyperplane density. As we will detail in the next Chapter, LDIS (2016), CDIS (2016), XLDIS (2018), and EGDIS (2020) [92] methods try to keep the densest instances in the training set. What differs from those methods is the way they define or capture density. **Spatial Hyperplane** category is composed of algorithms that divide the feature space with separating hyperplanes and perform sampling in each defined sub-division. The PSDSP [20] (2018) algorithm is a representative method of this category. Lastly, **Clustering-based** uses clustering techniques to aggregate instances and, later, perform a sampling within each cluster. The most recent and state-of-the-art is the Curious IS (CIS) [100] method (2022).

In the next section, we provide detailed information about the aforementioned IS approaches considered in our experimentation evaluation for the ATC context.

2.4 Instance Selection Methods Details

As previously mentioned, in [56], the authors proposed five categories to classify the IS algorithms according to their paradigm. We extended that taxonomy by proposing three new categories to contemplate recently proposed methods. In Table 2.2, we summarize all the considered IS methods, providing a concise description and the time complexity of each one, which we detail in the following paragraphs.

Condensed Nearest Neighbor (**CNN**) [62] is one of the most referenced approaches within the Condensation category. CNN starts from a solution set S containing a random instance of each class. Next, it iteratively predicts the class for each instance x in the original set of instances T (leveraging the K-Nearest Neighbors classifier). Finally, it includes in S the misclassified instances. CNN's authors consider the instances close to the classification boundary as the most representative ones. As these instances are more challenging to classify due to the diversity usually present in these areas, CNN's estimator considers, in each iteration, only instances present in S. This tends to promote the selection of more representative instances close to the boundary´s frontiers. The time

Method (Brief Description)	**Complexity**	**Proposed Application**
CNN [62] includes misclassified instances (leveraging KNN classifier) in the solution set S.	$O(n^3)$	2-D uniform dist
ENN [143] removes incorrectly classified instances from the solution set S.	$O(n^2)$	2-D uniform dist
IB3 [2] is based on a "wait and see" strategy choosing the best instances from the records.	$O(n^2 \log n)$	Tabular data
Drop3 [142] filters noise and removes instances further away from the decision boundary.	$O(n^3)$	Tabular data
ICF [12] is an association set approach based on concepts of reachability and coverage.	$O(n^2)$	Tabular data
LSSm [79] chooses instances based on their usefulness - influence over other instances.	$O(n^2)$	Tabular data
LSBo [79] inserts the instances in the solution sorted by the cardinality of the local set.	$O(n^2)$	Tabular data
LDIS [18] assumes as representative instances those with higher density.	$O(\sum_{l \in L} \|c(l)\|^2)$	Tabular data
CDIS [19] also keeps denser instances, but adopts another density concept.	$O(\sum_{l \in L} \|c(l)\|^2)$	Tabular data
XLDIS [17] extends LDIS adopting the local density ordering concept.	$O(\sum_{l \in L} \|c(l)\|^2)$	Tabular data
PSDSP [20] retrieves the centroid instances of a set of predefined hyperplane partitions.	$O(n)$	Tabular data
EGDIS [92] is a global density-based IS approach based on an irrelevance function.	$O(n^2)$	Tabular data
CIS [100] is a clustering-based approach that adopts curiosity and intrinsic reward concepts.	$O(n^3)$	Tabular data

Table 2.2: Summary of IS Methods.

complexity of CNN is $O(n^3)$, where n is the size of the original set.

The most traditional Edition method is the Edited Nearest Neighbor (**ENN**) [143]. ENN starts by inserting all instances of the original set T into the solution set S. Next, it uses the K-Nearest Neighbors classifier to iteratively classify all $x \in S$ (considering the set $\{S - \{x\}\}$ as possible neighbors). Finally, it removes incorrectly classified instances from S. The time complexity of ENN is $O(n^2)$.

Hybrid algorithms attempt to combine the condensation and editing paradigms, with the best examples being the IB1, IB2, IB3 [2] methods, DROP1-5 [142], ICF [12] and LSBo [79]. Likewise the condensation methods, IB1 starts with an empty solution set S, then finds the most similar instance y for each sample x present in the original set T. If the distance $d(x, y)$ is greater than a given threshold, it includes x in the solution set S. IB2 only inserts the erroneously classified instances into the solution set S, verifying whether the class of both instances x and y are the same. It includes x in the solution set S when it is not. The objective of IB2 is to find and insert in S instances closest to the decision boundary. Finally, **IB3** is the direct extension of IB2 – which selects and stores only the wrongly classified instances. However, IB3 is based on a "wait and see" strategy choosing the instances that generated the best classifiers given the selected records. The time complexity of IB3 algorithm is $O(n^2 \log(n))$.

The Decremental Reduction Optimization Procedure (**DROP1-5** [142]) approach is an ordering and filtering approach following N rules. DropN is a family of five different algorithms. This group of methods is defined through a set of decremental instance reduction procedures. The first rule (Drop1) removes instances that do not impact the model's generalization. It starts with S=T and, in a trial-and-error way, removes instances that do not degenerate the model's accuracy. This reduction rule is known to be effective for removing many instances, also tending to remove noise. Drop2 removes instances considering the original set T, checking whether removing an instance x could impact the classification of its neighbors in the set T. The main idea is to order the instances to minimize the impact on the neighborhood-based classification. Drop2 first removes the furthest instances for its nearest enemy (another class instance). The goal is to remove in-

stances further away from the decision boundary. **Drop3** consists of adding noise-filtering and then applying Drop2. The filtering is similar to ENN, removing any misclassified instances from the solution set S. There are two other rules (Drop4 and Drop5) based on more rigorous noise filters and the selection of decision boundaries. Since **Drop3** presents the best trade-off between reduction and accuracy [100, 92, 79], we will focus our studies on it. The time complexity of DROP3 algorithm is $O(n^3)$.

The ICF [12] is an association set-based approach based on concepts of reachability and coverage. Coverage of an instance x is the set of instances $y \in T$ such that the distance $d(x, y)$ is less than the distance from x to its nearest neighbor of another class (i.e., nearest enemy $ne(x)$) as defined in Equation 2.1.

$$Coverage(x) = \{y | d(x, y) < d(x, ne(x))\} \tag{2.1}$$

In turn, the reachability of an instance x is the set of instances y that have x in their Coverage set as stated in Equation 2.2.

$$Reachability(x) = \{y | x \in Coverage(y)\} \tag{2.2}$$

In summary, reachability regards the neighborhood of the instance, while coverage regards the associations of an instance. Consequently, ICF maintains instances that can classify others without explicitly maintaining them in the training set. In practice, this is done iteratively by discarding instances whose reachability is greater than coverage. The time complexity of ICF is $O(n^2)$.

In [79], the authors proposed three algorithms: Local Set-based Centroids Selector (**LSCo**), Local Set-based Smoother (**LSSm**) and Local Set Border Selector (**LSBo**), highlighting the results presented by the last two. While LSSm is considered an editing method and LSBo is a hybrid method, both leverage the concept of local set (LS). By definition, an LS is a set of instances contained in a sub-region of the feature space hyperplane, such that all instances that make up the LS are of the same class. In other words, considering an instance x, LS(x) can be defined as the set of instances y such that the **euclidean distance** between x and y is less than the euclidean distance between x and its nearest neighbor of another class (a.k.a. the nearest enemy of x – ne(x)). LSSm and LSBo also depend on three other definitions: usefulness (Equation 2.3), harmfulness (Equation 2.4), and LS cardinality (LSC) (Equation 2.5). Given an instance, e, usefulness $u(e)$ is the number of instances with e belonging to their LS. Harmfulness $h(e)$ is the number of instances that have e as their closest neighbor to another class ($ne(c)$). Finally, LSC is the number of instances belonging to the LS of e (LS(e)).

$$u(e) = |x | e \in LS(x)| \tag{2.3}$$

$$h(e) = |x | ne(x) = e| \tag{2.4}$$

$$LSC(e) = |LS(e)| \tag{2.5}$$

In LSSM the set S is composed of instances that have $u(e) > h(e)$. An instance e with high usefulness has importance/influence for many other instances. Consequently, e must belong to the solution set S. The time complexity of LSSm is $O(n^2)$. In turn, LSBo starts with noise removal by applying LSSM. Next, it calculates the local sets and orders the instances according to their LSC. Finally, inserts e into S if there is no intersection between e' local set and S. Since decision boundary instances will be computed and inserted first into the set S, these instances (e) will enable the correct (further) classification of the instances belonging to its LS. Like LSSm, the time complexity of LSBo is $O(n^2)$.

Unlike LSSm and LSBo, whose objective is to keep the instances present on the decision boundary, the **Density** approaches try to keep the instances that are present in denser regions. In [18], the authors proposed the Local Density-based IS (**LDIS**) method. LDIS analyzes each data class separately. LSSm and LSBo perform a global search in the dataset. In contrast, LDIS performs a local search by class, assuming as representative instances those with higher density (as defined in Equation 2.6). Consequently, the runtime complexity becomes $O(\sum_{l \in L} |c(l)|^2)$, where l is a specific class belonging to the set of classes L and $|c(l)|$ is the number of instances belonging to class l. In mode details:

$$Dens(x, P) = -\frac{1}{|P|} \sum_{y \in P} d(x, y) \tag{2.6}$$

where x is an arbitrary instance, P is the set of instances of the class to which x and y belong, d is a measure of distance. LDIS iterates over k instances y closest to x, inserting x into S if $Dens(x, c(l)) > Dens(y, c(l))$. This property guarantees that if an arbitrary instance x is denser than all its neighbors, it will be present in the solution set S.

Based on LDIS, two extensions emerged: Central Density-based IS (**CDIS**) [19] and eXtended Local Density-based IS (XLDIS) [17]. CDIS also uses the idea of keeping denser instances, evaluating them separately by class (*local search strategy*). However, it adopts another definition of density, as pointed out in Equation 2.7.

$$density(x, P) = \frac{\sum_{y \in P} \frac{1}{1+d(x,y)}}{1 + d(x, centroid(pkn(x, k)))} \tag{2.7}$$

where x is an arbitrary instance, and P is the set of class instances to which x and y belong. The numerator represents the multiplicative inverse of the x distance for every y of the same class. The denominator corresponds to the distance between x and the centroid of x's class. In this way, CDIS selects the densest instances closest to the centroid of each class. By using a local search strategy by class, the CDIS has time complexity proportional to $O(\sum_{l \in L} |c(l)|^2)$, where l is a specific class belonging to the set of classes L and $|c(l)|$ is the number of instances belonging to class l.

The *eXtended Local Density-based IS* (**XLDIS**) adopts the exact definition of density presented in Equation 2.6. Additionally, it introduces the definition of *local density ordering* (LDO) as the order of the instance x in the set $c(l(x))$ (i.e., the class it belongs to) according to its local density. The LDO defines the analysis order of the instances since the local density is directly related to how representative the instance x is to its neighbors. In short, XLDIS inserts x in S when x has the largest LDO among its partials k-neighborhood. The time complexity of XLDIS is $O(\sum_{l \in L} |c(l)|^2)$.

Most density techniques are based on the concept of local density – a function that evaluates an instance x by considering examples from the same class of x, which might lead to both reduction and accuracy improvements. However, this concept has some limitations. As these algorithms have a only local view of the dataset (locally by class), both reduction and effectiveness can be limited to the algorithm knowledge of the specific class. To address these limitations, the authors in [92] propose two global density-based IS algorithms called Global Density-based IS (GDIS) and Enhanced Global Density-based IS (**EGDIS**). The GDIS algorithm uses the relevance function to assess each instance's importance. In summary, the number of neighbors from the same class of an instance x determines the relevance of that instance. In the analyzed data tabular context, the GDIS algorithm achieves good classification accuracy values but with a decrease in reduction rate. EGDIS aims to address this issue using another function called the irrelevance function. This function determines the number of neighbors from another class. This modification improves the results by enhancing both the reduction rate and effectiveness. Since EGDIS presents the best trade-off between reduction and accuracy, we will focus our studies on it. The time complexity of EGDIS algorithm is $O(n^2)$).

Spatial HyperPlane algorithms divide the hyperplane space of the features to later choose representative instances of each subspace. In [20] the authors proposed the Prototype Selection method based on Dense Spatial Partitions (**PSDSP**), which also uses a local search strategy by class. First, PSDSP separates instances by class, and for each class divides the hyperplane into n partitions of the same size. Then, PSDSP retrieves the centroid for each hyperplane (in descending order by the number of instances) and inserts it into S. The time complexity of PSDSP algorithm is linear ($O(n)$).

Finally, the Curious IS (**CIS**) [100] is a clustering-based strategy that incorporates the notions of intrinsic reward and curiosity. CIS starts by clustering the instances, where each cluster is considered a system state. Starting without any cluster in the solution, the reward agent selects a new cluster of instances in each loop episode to join the already selected clusters (state). The intrinsic reward is proportional to the decrease in the learner's prediction error. Ultimately, the algorithm's output is a matrix representing the trade-off between model improvement and the selected data size. The time complexity of CIS method is $O(n^3)$.

2.5 Summary

In this chapter, we presented a critical analysis of the state-of-the-art proposals in the Instance Selection field. The objective of this evaluation was to comprehensively assess the most relevant works related to IS strategies applied in different scenarios. In particular, we focused on experimentally-oriented studies, that is, studies that have strong experimental and empirical components to validate their conclusions.

The results of our searches and analyses reinforced our perception that IS methods are almost exclusively applied to tabular structured data — and their application in NLP tasks is rare, which is odd since this is one of the areas that could benefit most from this type of method. In sum, from the 100 considered IS papers, 92 of them considered just tabular data. We propose to investigate the use of IS methods along with ATC models, which have been highly popular in applications as diverse as the detection of fake news and hate speech, sentiment analysis, revision of product characteristics, inferring opinions, and assessing the satisfaction of products and services, among many others.

We also proposed a new taxonomy of IS strategies by extending the one proposed in [56]. As mentioned, this prior IS methods taxonomy was proposed in a different context, assessing different dataset types (small tabular ones) and other learning techniques (e.g., deep learning neural network techniques were not considered in the envisioned scenarios).

Last, we also provided detailed information about the thirteen IS approaches evidenced in our critical analysis that we will consider in our experimentation evaluation for the ATC context. Therefore, in the next chapter, we provide a complete experimental evaluation comparing the IS methods detailed above, combining them with state-of-the-art automatic text classification strategies. Our objective with this set of experiments is to answer the first posed research question (RQ1).

Chapter 3

A Comparative Survey of Instance Selection Methods applied to NonNeural and Transformer-Based Text Classification

In this chapter, we propose to assess the tradeoff among reduction, efficiency, and effectiveness of these 13 most representative traditional IS methods (Section 2.4) applied to the ATC task using large and varied datasets. It is essential to notice that the selected IS methods presented in the previous chapter have been tested only with small structured tabular datasets (such as those from the UCI repository). Regarding ATC methods, we considered the current SOTA in the text classification field: the transformer-based architectures, such as BERT, XLNet, RoBERTa, and others. These methods have a high cost in terms of computational resources, mainly when dealing with large labeled training data. Therefore, they constitute an ideal scenario for the application of IS techniques.

In more detail, we can divide the use of deep learning networks into two phases. In the first phase, there is a massive training where the weights of deep learning networks are estimated on unsupervised tasks (e.g., masked language model and next word prediction) over a huge unlabeled dataset. This phase usually requires huge amounts of training data and massive computational power, meaning that only a few corporations are usually capable of performing such tasks. The most common use of these pre-trained network models encompasses a second stage in which a fine-tuning step is necessary to adjust the model to a different and specific domain (a.k.a., domain transfer) as we apply the model to a task potentially different from the initially proposed one. Fine-tuning these networks requires fewer examples than learning from scratch. As the methods tested in the present work are supervised and, therefore, demand the class label of each document, our focus in this work is to study the effect of the application of IS methods on the fine-tuning stage (See Section 3.3.3). This is also the most common task performed by most researchers and practitioners when applying deep network models to real-world ATC tasks.

As with most NLP tasks, ATC is directly impacted by the availability of large training data and the new computationally expensive approaches. In our investigation, we considered the current state-of-the-art text classification methods: the Transformer-based architectures, such as BERT, XLNet, RoBERTa, and others. These methods have high costs in terms of computational resources, mainly when dealing with large training data. As such, they constitute an ideal scenario for applying IS techniques. Last, but not least important, we should emphasize that our work, as far as we know, is the first to apply IS as a preprocessing step before using transformer-based architectures in the ATC context. This contribution is realized using an experimental study whose rigor and magnitude (seven transformer methods and thirteen IS approaches) have not yet been reported in the literature on IS.

3.1 Comparative Scenario and Experimental Setup

This section introduces the experimental setup, which includes: (i) the used datasets; (ii) the considered text classification and IS methods; (iii) the data representation and preprocessing techniques; and (iv) the evaluation metrics and experimental protocol.

3.1.1 Datasets

To evaluate IS methods, we consider **nineteen** datasets in two types of text classification task: i) **topic classification**; and ii) **sentiment classification** (a.k.a. polarity detection). Regarding the division into Sentiment Analysis and Topic classification, most studies on text classification organize their experiments into these two broad categories [80, 98]: associating a sentiment (polarity) or a subject (topic) with a piece of text. Classical references show that these two subtasks constitute the main text classification tasks [121, 5, 80, 14, 34]. Within these two broad text classification tasks, the considered datasets represent several domains and applications. For the topic classification task, we consider domains as diverse as web pages categories (WebKB), newsgroups (20NG), scientific papers in computer science (ACM and DBLP), academic articles/journals categorization (Web Of Science), medical documents (OHSUMED), books genres (Books),

and question subject classification (TREC). For the sentiment classification task, we considered the domains of movie reviews (MR), product reviews (yelp_reviews), opinions and comments about businesses, news comments (vader_nyt), movie sentiment polarity, binary subjective or objective classification (Subj), and opinion polarity detection (MPQA).

Justifications on the Datasets' Representativeness and the Tasks All these datasets [1] have been widely used as benchmarks by most works in the text classification field [80, 98]. The large majority of the works in the literature on text classification organize their experiments into these two broad categories: Sentiment Analysis and Topic classification. Indeed, classical references show that these two subtasks constitute the main text classification tasks [121, 5, 80, 14, 34].

Indeed, in a recent survey in the field [80], considering traditional and deep learning approaches, the authors evaluated 38 datasets used in 51 different works. The datasets evaluated in our work are present in more than half of them (specifically, in 33 of these works). In that work, all 38 datasets are categorized either for sentiment analysis (11) or topic classification (27). The topic classification category is further subdivided into four subclasses: (i) News Classification (e.g., 20NG dataset); (ii) Topic Labeling (e.g., OHSUMED); (iii) Question Answering (e.g., TREC); (iv) multi-label (e.g., Reuters[2]); Note that, even considering this topic classification category subdivision, in our work we consider representative datasets of each subclass.

Another recent work [98] consists of a summary of more than 40 popular datasets widely used for text classification. In that work, the datasets are divided into five categories. The first one is associated with Sentiment Analysis containing six datasets (of which five are used in our work). Next, there are three classes associated with topic classification, being News Classification (12), topic labeling (11), and question answering (8). Finally, a class associated with natural language inference (NLI) contains seven datasets. NLI is used to determine whether the meaning of one text can be inferred from another, a task involving pairs of sentences. NLI is outside of the scope of our work which focuses on multi-class and single-label tasks for textual documents. Thus, this last class of datasets is the only one not covered by our work.

In sum, these surveys confirm that sentiment analysis and topic classification are the two most important tasks in the field. Regarding representativeness, considering both works, 15 datasets among the 19 we used in our experimentation are discussed in at least one of the surveys mentioned above.

As detailed in Table 3.1, we can observe diversity in many aspects of these datasets, in terms of size, dimensionality (i.e., number of terms), the number of classes, density (the average number of words per document) and class distribution. These datasets have

[1] See Appendix A for further information about the datasets

[2] In our case, we adopted the single-label version of the Reuters dataset.

different levels of skewness, ranging from completely balanced (20NG) to completely unbalanced, such as the case of Reuters90, where the minority class has only 2 documents.

Task	Dataset	Size	Dim.	Class Distribution					Density	Skewness
				# Classes	Minor	Median	Mean	Major		
Topic	DBLP	38,128	28,131	10	1,414	3,590	3,812	9,746	141	Imbalanced
	Books	33,594	46,382	8	1,226	4,534	4,199	4,934	269	Imbalanced
	ACM	24,897	48,867	11	63	2,041	2,263	6,562	65	Imbalanced
	20NG	18,846	97,401	20	628	984	942	999	96	Balanced
	OHSUMED	18,302	31,951	23	56	592	795	2,876	154	Imbalanced
	Reuters90	13,327	27,302	90	2	29	148	3,964	171	Extremely Imbalanced
	WOS-11967	11,967	25,567	33	262	371	362	449	195	Balanced
	WebKB	8,199	23,047	7	137	926	1,171	3,705	209	Imbalanced
	TREC	5,952	3,032	6	95	1,148	992	1,344	10	Imbalanced
	WOS-5736	5,736	18,031	11	380	426	521	750	201	Balanced
Sentiment	SST1	11,855	9,015	5	1,510	2,242	2,371	3,140	19	Balanced
	pang_movie	10,662	17,290	2	5,331	5,331	5,331	5,331	21	Balanced
	Movie Review	10,662	9,070	2	5,331	5,331	5,331	5,331	21	Balanced
	vader_movie	10,568	16,827	2	5,242	5,284	5,284	5,326	19	Balanced
	MPQA	10,606	2,643	2	3,312	5,303	5,303	7,294	3	Imbalanced
	Subj	10,000	10,151	2	5,000	5,000	5,000	5,000	24	Balanced
	SST2	9,613	7,866	2	4,650	4,806	4,806	4,963	19	Balanced
	yelp_reviews	5,000	23,631	2	2,500	2,500	2,500	2,500	132	Balanced
	vader_nyt	4,946	12,004	2	2,204	2,473	2,473	2,742	18	Balanced

Table 3.1: Datasets Statistics

3.1.2 Data Representation and Preprocessing

We have several options to use as (vectorial) representation input for the IS methods, which include TF-IDF weighting-scheme-based representation, static embeddings (Word2vec, GloVe, fastText), and contextual embeddings (resulting from a zero-shot or fine-tuning model).

For several reasons, summarized below, and given the scope of this dissertation, we have chosen to use the TF-IDF weighting-scheme-based representation.[3] First of all, TF-IDF is a straightforward, easy-to-understand, efficient, and popular representation in the ATC realm.

Second, according to [31], the use of static embeddings such as FastText[68], FisherVector[77]; and PTE[129] leads to significant effectiveness losses when compared to standard TFIDF+SVM in several of the datasets we use in our study. Furthermore, exploring static embeddings can incur significant increases in computational cost – between 1.5x and 31.1x slower than the traditional TFIDF representation. Therefore TF-IDF is a better choice regarding a trade-off effectiveness-cost.

Fine-Tuned Contextual embeddings (e.g. BERT-based) could be a choice as they leverage most state-of-the-art models in many natural language processing tasks. However, despite the potential benefits in terms of effectiveness [151], in the context of IS, fine-tuning before applying the IS methods for selection would be unfeasible given the

[3]For a detailed explanation about the alternatives for the IS Input representation, please see Appendix C.

usual costs of this step, which could be even higher than the selection itself. In other words, fine-tuning to select and train again does not make much sense.

Finally, a less complex and computationally less expensive option for using contextual embeddings along with IS would to exploit just the pre-trained models without any tuning (aka, *zero-shot approach*). In Appendix C, we evaluate this option coming to the conclusion that using pre-trained zero-shot embeddings without further investigations is either inefficient or ineffective or both. Overall, this exercise (see the details in the Appendix) revealed that using contextual embeddings along with IS methods is not trivial and will require further research considering the current state-of-the-art of both fields.

In sum, considering all aforementioned reasons, the **TF-IDF** representation is used as input to all IS methods and for MetaFeatures generation. As a pre-processing step, before creating the TFIDF matrix, we adopted the following steps: i. we removed stopwords using the standard list from the scikit-learn library [107] (version 0.23.2); and ii. we only kept features that appear in at least two documents.

In this chapter, we also consider MetaFeatures[16] for the sake of representation combined with the SVM classifier. Strategies based on **MetaFeatures (MFs)** extract information from other more basic features (such as TFIDF) aiming at improving the feature space based on the main assumption that close documents tend to belong to the same class [97]. Indeed, these strategies work by enriching the input/representation space and are combined as metaclassifiers' input. The MFs we exploit here are the most effective ones according to [15]. We adopt the combined similarity scores (cosine and $l2$) between a document and each category centroid as meta-features that exploit global information, and the similarity between a document and its neighbors from each category as meta-features that exploit local information. Such meta-features evaluate the proportion of correctly classified (using the SVM classifier) neighbors of a target document and the discrepancy between the classification of a target document and its neighbors. Our implementation[4] of the MF approach was obtained from the authors of the original work.

Finally, as mentioned before, our study also considers end-to-end neural networks (E2E). Thus, the input considered is a raw document representation (original text) for the classification methods based on deep learning (BERT, XLNet, RoBERTA, GPT-2, DistilBERT, AlBERT, and BART). As illustrated in Figure 3.1, we first split the dataset into k train-test subsets employing the stratified k-fold cross-validation methodology [67]. The experiments in the smaller datasets were executed using k=10-fold partition, while for the larger ones, we adopt 5 folds due to the cost of the procedure. Subsequently, for each fold, we construct the TFIDF[5] matrix representation of the documents for the IS stage. After selecting the documents, the ATC models undergo method-dependent preprocessing and receive the respective raw documents as input (and not the TFIDF representation).

[4]Available in https://gitlab.com/waashk/extended-pipeline

[5]The IDF is calculated based on the training only preventing any information lacking.

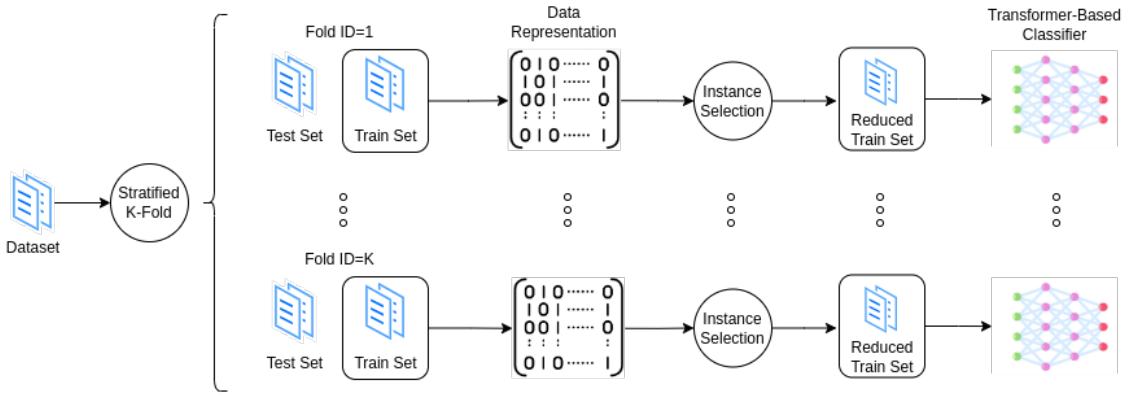


Figure 3.1: Data Representation and Preprocessing Procedure

3.1.3 Text Classification Methods

We consider the same approaches used in [34]. In that work, the authors performed a thorough and rigorous comparative study of the cost-effectiveness of neural and non-neural approaches and representations for automatic text classification (ATC), including (i) classic approaches for data representation with traditional classifiers such as TFIDF weighting with SVM; (ii) the combination of representations based on Metafeatures with an SVM classifier; and (iii) end-to-end neural network approaches such as Convolutional NNs, Long Short Term Memory NNs (LSTM), Graph Convolutional Networks (GCN) as well as more recent approaches such as BERT and XLNet. As aforementioned, in addition to studying the behavior of IS techniques, our objective in this work is to extend the results presented in [34], adding recent classification approaches proposed in the literature, most notably those based on Transformer architectures: RoBERTa, GPT, DistilBERT, ALBERT, and BART. As far as we know, evaluating traditional and recent IS strategies in the context of automatic text classification, mainly the Transformer-based architectures, is a major novelty of our work as well as one of its main contributions.

Method-Specific Parameter Tuning: All parameters for the non-neural methods were defined with grid-search, using cross-validation in the training set and we report the effectiveness of the algorithms in the test partition. For MetaFeatures (MFs), we experimented with different neighborhood sizes (parameter k), varying from 10 to 50 (with linear increments of 5), to choose the one with the highest effectiveness in the validation set. We used the LIBLINEAR [48] implementation of the SVM method to generate both TFIDF-based and MetaFeatures (MF) based classifiers. LIBLINEAR is still one of the best SVM implementations for text classifiers, capable of handling high dimensions (TFIDF) and low dimensions (MF). The SVM regularization parameter was chosen from eleven values from 2^{-5} to 2^{15} using the validation sets.

For Neural Network methods, it is impossible to use grid-search with cross-validation for all the hyperparameters, given the large number that must be tuned. Therefore, to define the best hyperparameters, we adopt the approach used in [34] and follow, as a general rule, the suggestions of the authors. Furthermore, since the max_len and batch_size hyperparameters directly impact issues of efficiency and effectiveness, we perform a GridSearch on these specified values. Table 3.2 contains the definition of fixed parameters and the values considered in GridSearch for the hyperparameters max_len and batch_size.

methods	**parameters**
BERT	initial learning rate: 5e-5
XLNet	batch size: [16,32]
RoBERTa	max_len: [150,256]
GPT2	patience: 5
DistilBERT	max_epochs: 20
AlBERT	weight_decay_rate: 0.01
BART	max_grad_norm: 1.0

Table 3.2: Parameters Tunning of the Transformers Neural Networks

As the literature [14, 32] suggests, we did not perform any pre-processing for classifiers based on neural network embeddings. The authors of [14] argue that simple tokenization works as well or even better than complex preprocessing methods. Furthermore, in preliminary tests, we noticed that these neural networks achieved similar or worse results when pre-processing methods were applied. Also, following the literature [34, 10], we perform the fine-tuning with all trainable layers (without any "frozen" layer). Instead, to port all neural networks based on embeddings to the text classification task, we include a fully connected layer (with a size equal to the number of classes of each dataset) at the end of each network, thus enabling the direct application in the text classification task.

3.1.4 IS Methods.

We consider in this chapter a set of 13 IS methods described in Section 2.4, namely: *Condensed Nearest Neighbor* (**CNN**); *Edited Nearest Neighbor* (**ENN**); *Iterative Case Filtering* (**ICF**); *Instance Based 3* (**IB3**); *Decremental Reduction Optimization Procedure* (**Drop3**); *Local Set-based Smoother* (**LSSm**); *Local Set Border Selector* (**LSBo**); *Local Density-based Instance Selection* (**LDIS**); *Central Density-based Instance Selection* (**CDIS**); *eXtended Local Density-based Instance Selection* (**XLDIS**); *Prototype Selection based on Dense Spatial Partitions* (**PSDSP**); *Enhanced Global Density-based Instance Selection* (**EGDIS**); and *Curious Instance Selection* (**CIS**).

Method-Specific Parameter Tuning: All parameters for the IS methods were defined with grid-search, using cross-validation in the training set during an initial empirical experiments round. Table 3.3 shows the range of parameter values for each IS method we evaluate. The best parameter in each range is marked in **bold**.

method	parameters
CNN LSSm LSBo	n_neighbors: [**1**, 3, 5, 10]
ENN Drop3 LDIS CDIS XLDIS EGDIS	n_neighbors: [1, **3**, 5, 10]
IB3	Confidence Acceptance: 0.9 Confidence Dropping: 0.7
PSDSP	n_neighbors: [1, 3, **5**, 10] p: [0.05, **0.1**, 0.2]
CIS	iterations: $100 * \|k_{cluster}\|$ learner: Decision Tree initial error: 0.5 discount factor: 0.01 epsilon: 0.9 to 0.1 (step decay) learning rate: 0.09 to 0.01 (step decay)

Table 3.3: Parameters of the IS methods

3.1.5 Evaluation Metrics and Experimental Protocol

We evaluated the IS methods concerning the capacity to reduce the training set, classification effectiveness, and training time. Experiments were executed on an Intel superscript registered Core i7-5820K with 6-Core and 12-Threads, running at 3.30GHz, 64Gb RAM, and a GeForce GTX TITAN X (12GB) and Ubuntu 19.04.

According with [79], reduction mean (Equation 3.1) is described as:

$$\overline{R} = \frac{\sum_{i=0}^{k} \frac{|T_i| - |S_i|}{|T_i|}}{k} \tag{3.1}$$

where T is the original training set, S is the solution set containing the instances selected by the IS method being evaluated, and k is the number of folds adopted in our experiments (10 folds).

We assessed classification effectiveness with Macro Averaged F1 (MacroF1) [127]. MacroF1 measures the classification effectiveness for each class, averaging them. In order to compute the F1 measure, the system-made decisions on the test set concerning a specific category must be divided into three groups: True Positives (TP), False Positives (FP), and False Negatives (FN), respectively. The terms positive and negative refer to the classifier's prediction, and the terms true and false refer to whether that prediction corresponds to the external judgment. The measure is described as:

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{3.2}$$

The experiments were executed using a 10-fold cross-validation procedure. To compare the average results on our cross-validation experiments, we assess the statistical significance employing the paired t-test with 95% confidence, which is strongly recommended over signed-rank tests for hypothesis testing on mean effectiveness and arguably robust to potential violations of the normality assumption in this context [133, 65]. To account for multiple tests, we adopt the Bonferroni correction [64].

We summarize our results by performing a fractional ranking analysis to sort the best IS methods and ATC algorithms. In fractional rank, items that compare equally receive the same ranking number, which is the mean of what they would have under ordinal rankings. For instance, the sorted ranking scores: 1, 2.5, 2.5, and 4, are obtained when comparing four methods with the second and third being statistically tied. The two methods with equal rankings (2.5) indicate that there is no significant statistical difference between them. In our scenario, we rank each method for each dataset based on the MacroF1.

We also assess the cost of each method in terms of the model construction time, aiming at analyzing the cost-effectiveness trade-offs for all methods. The metric is the overall time in seconds (average of 10 folds). The Speedup (Equation 3.3) is measured as follows:

$$S = \frac{T_{wo}}{T_w} \tag{3.3}$$

where T_{wo} is the complete execution time without the IS phase, and T_w is the complete model construction time using the IS method.

3.2 Preliminary Question: What is the best (most effective) classification method/ representation for each of the considered datasets?

In [34], we presented a study of several classifiers and representations. Since the publication of [34], other ATC approaches have emerged or demonstrated to be highly effective for the task. As we propose to answer the RQ1 based on a tripod that considers effectiveness as one of the base pillars, we believe that a precise and up-to-date response to this preliminary question is necessary. Therefore, we decided to re-establish the SOTA in each dataset so that the results are as strong as possible and do not weaken this pillar.

Indeed, the SOTA in the ATC area changes very quickly, as the results we obtain when answering this preliminary question demonstrate, the best methods for each dataset are not the same ones pointed out in the aforementioned recent comparative study in the area [34]. Last, and more importantly, as also shown in [34], the results from the literature are difficult to compare due to inconsistencies in experimental procedures and a lack of statistical rigor when reporting results. By revisiting the question using rigorous and uniform experimental and statistical procedures across all datasets, we establish a strong foundation for the present study.

Thus, our first contribution in this Ph.D dissertation is complementary to [34], considering new datasets, classifiers, and representations. More specifically, we considered 15 new datasets and included, in addition to the results of the best methods found in that work (TFIDF+SVM, MF+SVM, BERT, and XLNet), five new end-to-end Transformer-Based classifiers (described in Appendix B) considered to be the state-of-the-art in several tasks, namely: RoBERTa, GPT2, DistilBERT, AlBERT, and BART. Besides greatly extending [34], this analysis aims to support the application of IS methods in the best possible scenario (top-best-ATC-method) for each of these datasets. The results of all these evaluations are shown in Table 3.4.

Task	Dataset	RoBERTa	BART	XLNET	BERT	DistilBert	Albert	MF+SVM	GPT2	TFIDF
Topic	DBLP	**81.4(0.5)** ●	**81.1(0.5)** ●	**81.4(0.6)** ●	**81.7(0.5)** ●	**81.0(0.6)** ●	77.3(1.0)	80.5(0.7)	78.9(0.8)	79.3(0.7)
	Books	87.2(0.6)	86.9(0.5)	87.3(0.4)	**89.5(0.2)** ▲	87.5(0.5)	84.6(0.8)	88.3(0.3)	85.4(0.7)	84.1(0.4)
	ACM	**70.3(1.4)** ●	**70.8(0.7)** ●	**69.9(0.9)** ●	**71.8(1.0)** ●	**70.1(1.0)** ●	66.2(1.9)	**70.3(1.0)** ●	67.6(1.2)	68.0(0.7)
	20NG	86.8(0.7)	87.4(0.9)	87.4(0.8)	85.4(0.5)	86.7(0.6)	76.9(1.2)	**90.7(0.6)** ▲	82.3(0.9)	89.1(0.7)
	OHSUMED	**77.8(1.2)** ●	**77.6(0.7)** ●	**77.6(1.0)** ●	**76.4(1.2)** ●	**76.2(0.7)** ●	66.1(4.8)	71.8(1.0)	74.5(0.8)	71.2(1.1)
	Reuters90	41.9(2.2)	42.2(2.1)	41.3(2.6)	40.2(2.8)	40.7(2.5)	41.0(2.6)	**48.4(2.6)** ▲	37.2(2.3)	31.9(3.2)
	WOS-11967	**86.8(0.4)** ●	**86.9(0.8)** ●	**87.0(0.7)** ●	85.5(0.7)	**86.0(0.7)** ●	76.8(1.1)	82.0(0.9)	81.5(0.9)	84.5(0.6)
	WebKB	**83.0(2.0)** ●	**83.0(1.7)** ●	**81.9(2.5)** ●	**83.2(2.1)** ●	**82.3(2.1)** ●	**80.3(1.4)** ●	71.6(2.4)	79.0(1.9)	72.9(2.1)
	TREC	**95.5(0.5)** ●	**95.5(0.8)** ●	**94.3(1.1)** ●	87.6(1.4)	**95.5(1.1)** ●	**93.5(1.4)** ●	67.4(1.5)	92.0(1.0)	68.3(2.0)
	WOS-5736	**90.5(0.9)** ●	**89.6(1.7)** ●	**90.2(0.9)** ●	**89.7(1.3)** ●	**89.2(0.9)** ●	86.7(1.3)	87.2(0.8)	83.8(0.5)	**90.4(0.7)** ●
Sentiment	SST1	**53.8(1.3)** ●	**52.8(1.0)** ●	**51.4(1.7)** ●	**51.6(1.2)** ●	48.9(1.1)	49.2(1.2)	28.2(0.7)	45.4(1.1)	29.6(0.8)
	pang_movie	**89.0(0.4)** ●	**88.1(0.5)** ●	**88.2(0.6)** ●	87.4(0.4)	85.2(0.6)	**82.9(4.2)** ●	33.4(0.1)	81.7(0.8)	77.0(1.0)
	MR	**89.0(0.7)** ●	**88.2(0.6)** ●	**86.4(3.3)** ●	**87.7(0.5)** ●	85.2(1.1)	84.9(1.2)	33.5(0.2)	81.6(0.8)	75.8(0.9)
	vader_movie	**91.3(0.5)** ●	**90.4(0.6)** ●	**90.5(0.4)** ●	88.2(0.7)	86.6(0.7)	85.4(1.6)	33.6(0.1)	85.0(0.5)	78.0(0.9)
	MPQA	**90.2(0.8)** ●	**90.1(0.7)** ●	88.6(0.5)	**89.1(0.7)** ●	88.5(0.6)	87.9(0.6)	76.9(0.6)	86.5(0.6)	78.3(0.7)
	Subj	**96.9(0.4)** ●	**96.8(0.4)** ●	**96.1(0.5)** ●	**97.0(0.3)** ●	**96.0(0.4)** ●	95.5(0.7)	90.0(0.7)	94.6(0.4)	89.1(0.6)
	SST2	**93.2(0.6)** ●	**92.8(0.5)** ●	92.1(0.4)	91.5(0.6)	89.6(0.5)	88.6(2.1)	79.2(0.8)	86.9(0.6)	79.0(0.7)
	yelp_reviews	**97.9(0.4)** ●	**97.5(0.4)** ●	**97.3(0.4)** ●	95.6(0.6)	95.6(0.6)	93.9(0.9)	33.5(0.2)	93.5(0.7)	94.7(0.8)
	vader_nyt	**85.3(0.6)** ●	**85.5(0.8)** ●	82.7(1.1)	80.7(0.9)	79.9(1.2)	76.9(1.8)	37.8(0.9)	74.9(1.8)	64.5(1.8)

Table 3.4: Results regarding the evaluation metric MacroF1. Legend: (a) ▲: the classification approach is superior to **all others; (b)** ●: the classification approach presents the highest result in terms of absolute values, but there are statistical ties with **other approaches; (c)** ●: the classification approach is statistical equivalent to the best approach (**marked with** ●) in dataset (line) considered.

We define as the **best approach** (by dataset), the one whose result of the metric (MacroF1) was the highest among all (in terms of absolute values). Therefore, an approach is considered the best for a dataset if it belongs to one of the following three cases: (i) its absolute value and the result are statistically superior to all other approaches ▲ (e.g., MF+SVM for two datasets: 20NG and Reuters90); (ii) it's absolute value is superior but statistically equivalent to other approaches ● (e.g., BERT in the WebKB dataset); and (iii) its confidence interval is the smallest in case of a tie in the highest absolute value with other approaches (e.g., RoBERTa and BART in the TREC dataset).

We observe that the results presented in Table 3.4 are consistent with those presented in previous works [34] for the evaluated classifiers (SVM, MF, BERT, and XLNet). For instance, here as in [34], the best classification approach for 20NG and Reuters90 is MF+SVM. This may be explained by the fact that MetaFeatures are based on intra and inter-class distances and, as such, they produce enriched semantic information that is not directly captured by Transformer-based methods. This benefits datasets with more classes. Indeed, Reuters90 has the largest number of classes while the 4th largest is 20NG. MetaFeatures also performed relatively well in the WOS-11967 (2nd largest) and OHSUMED (3rd largest) datasets.

We observe, also, the insertion of the BART model in our analyses produced a new state-of-the-art benchmark result for ACM and WebKB datasets, with gains ranging from 0.37 to 4.41 percentage points when comparing with previous results of [34], where

the best approaches for these datasets were MF+SVM and XLNet, respectively.

Our extended comparative investigation reveals that considering statistical ties, in 17 datasets, the best classifiers[6] (out of nineteen) are constituted by a Transformer-based method, more specifically: RoBERTa (10), BERT (5), BART (1) and XLNet (1). Thus, we can note that among the methods considered in this work, primarily deep learning-based methods (transformers) are among the best in several domains. Considering only the sentiment datasets, the RoBERTa model obtained the best result in 78% of the datasets. Regarding the behavior of RoBERTa, the authors of the method demonstrated in [85] that, in fact, the increase in the amount of BERT training data and the increase in the training batch (main proposals of ROBERTa), were able to optimize the effectiveness of the proposed method, mainly in for sentiment datasets. Recent work [1] also demonstrates that RoBERTa (and derivatives of this model) are considered state-of-the-art in several sentiment analysis tasks, such as product reviews (yelp_reviews and amazon_reviews), and movie reviews (IMDB and SST).

Following, BERT was the best in five datasets (four topics and one sentiment: WebKB, ACM, Books, DBLP vader_nyt). Furthermore, considering only topic classification, we note that the best results, in terms of the number of datasets, are obtained by the BERT approach. Among the non-Transformers, MF+SVM obtained the best (unique) result in two datasets (20NG and Reuters90), which shows that it still is a very competitive method, especially for topic classification. Finally, BART and XLNet were better in only one dataset each (WOS-11967 and vader_nyt, respectively). In short, in Table 3.5 we provide complete, comparable, and statistically tested results summarizing the best classification method for each of the datasets studied here.

Task	Method	Dataset
Sentiment	RoBERTa	SST1
		pang_movie
		MR
		vader_movie
		MPQA
		SST2
		yelp_reviews
	BERT	Subj
	BART	vader_nyt

Task	Method	Dataset
Topic	BERT	DBLP
		Books
		ACM
		WebKB
	RoBERTa	OHSUMED
		TREC
		WOS-5736
	MetaFeatures + SVM	20NG
		Reuters90
	XLNet	WOS-11967

Table 3.5: Best ATC Approach by Dataset

It is worth explain the reasons for some differences regarding BERT´s results when compared to those obtained in [34]: (1) the BERT implementation used in [34] was in

[6]Number of times an approach was marked as ▲ or ●.

MXNet [26] architecture. Certain approximations made to optimize MXNet caused harm in the result achieved by this method. Here, we adopted an implementation in PyTorch [66]; (2) over the years, we have performed a more extensive search for BERT hyper-parameters, arriving at a configuration that benefited it in terms of effectiveness. In summary, both the dataset and the partitions (train-val-test) remain the same, but we now have a better implementation and configuration of the BERT algorithm.

Following, we present the results of each analyzed aspect of the tripod (reduction-effectiveness-efficiency) incrementally. First, we show the reduction potential (Table 3.6), then the effect of the reduction in effectiveness (Table 3.7 and Table 3.8), and finally, the effect in terms of efficiency (Table 3.9). We incrementally present the results to analyze each component of the tripod both in isolation and in conjunction.

3.3 Experimental Results - Analyses

In this section, we present the results of applying traditional IS methods in the context of Automatic Text Classification considering the set of the first raised research question (RQ1): *What is the impact of applying traditional IS methods in the ATC context regarding the posed constraints?* In order to conduct a thorough evaluation of each constraint, we have divided this RQ into three incremental sub-questions (RQ1.1–RQ1.3).

3.3.1 RQ1.1. Are there IS methods capable of reducing the training set while keeping classifier effectiveness for each investigated scenario?

We now enter the main focus of this work – the study of IS methods applied to ATC. In theory, IS methods should remove noisy and/or redundant instances as they aim to select the most representative instances and reduce the total time while trying to improve effectiveness. Some works have studied the behavior of IS mainly applied to tabular and low-dimensional data. The study of IS in the context of ATC introduces several challenges such as: (i) high dimensionality and sparseness; (ii) larger datasets (usually much larger than those explored in previous studies), (iii) noise and ambiguity in the text of the documents. In this context, a question that naturally arises is: “Are the results achieved by the IS methods in the previously studied contexts of tabular data extendable

to ATC?". We delve into this question next.

Departing from the premise that the construction time of a machine learning model is intrinsically associated with the amount of training data, we analyze the impact of applying IS approaches in terms of training set reduction and effectiveness. The reduction can be briefly described as the ability of a method to remove instances from the training set. Several works deal with the selection of important features to represent a model. In this work, we are focused on the selection of important instances. In practice, we can think of reduction as removing rows from a TFIDF representation matrix or removing documents from a textual dataset. Note that reducing the training set and maintaining effectiveness are conflicting goals. In Table 3.6, we present the results regarding the average reduction rate (Fold Average of a 10-Fold CV procedure) achieved by each selection method. A green color scale for each line (dataset), accompanied by the respective value, is shown in the Table. The darker a cell, the larger the reduction achieved by the corresponding method in the respective dataset.

task	dataset	CNN	ENN	ICF	IB3	Drop3	LSSm	LSBo	LDIS	CDIS	XLDIS	PSDSP	EGDIS	CIS	Average
Topic	DBLP	52.4%	24.6%	83.5%	40.0%	80.2%	17.4%	72.8%	85.6%	90.3%	87.8%	90.0%	62.0%	82.0%	66.8%
	Books	32.1%	24.9%	67.3%	15.0%	73.4%	8.8%	63.7%	85.1%	88.2%	87.5%	90.0%	62.0%	80.0%	59.8%
	ACM	47.1%	30.9%	78.3%	56.0%	75.8%	19.0%	67.7%	84.0%	88.0%	88.4%	90.0%	55.0%	46.0%	63.6%
	20NG	27.9%	76.1%	82.8%	5.0%	84.3%	0.5%	23.2%	94.3%	94.1%	95.1%	90.0%	68.0%	50.0%	60.9%
	OHSUMED	45.5%	31.8%	78.8%	53.0%	75.7%	21.9%	69.8%	88.3%	90.1%	89.8%	90.0%	57.0%	80.0%	67.1%
	Reuters90	50.7%	38.0%	86.9%	1.0%	84.7%	28.4%	76.9%	90.1%	88.6%	92.1%	90.0%	54.0%	67.0%	65.3%
	WOS-11967	45.4%	31.8%	78.7%	54.0%	76.7%	22.1%	68.4%	91.7%	91.3%	92.5%	90.0%	57.0%	77.0%	67.4%
	WebKB	42.9%	33.4%	77.8%	52.0%	78.9%	24.1%	71.1%	88.7%	90.1%	90.7%	90.0%	53.0%	57.0%	65.4%
	TREC	31.3%	61.4%	71.9%	41.0%	85.0%	18.4%	37.8%	88.8%	88.4%	97.0%	90.0%	39.0%	22.0%	59.4%
	WOS-5736	50.4%	28.2%	79.8%	59.0%	77.5%	20.1%	70.9%	90.1%	90.9%	91.4%	90.0%	62.0%	69.0%	67.6%
Sentiment	SST1	18.9%	74.7%	92.5%	31.0%	98.7%	5.7%	7.7%	99.3%	99.1%	99.8%	90.0%	20.0%	60.0%	61.3%
	pang_movie	46.8%	29.8%	72.2%	66.0%	73.0%	18.8%	63.5%	80.1%	94.4%	83.4%	90.0%	63.0%	77.0%	66.0%
	MR	46.7%	49.5%	80.7%	67.0%	99.2%	3.3%	48.8%	99.0%	99.3%	99.1%	90.0%	63.0%	58.0%	69.5%
	vader_movie	47.2%	28.9%	71.8%	67.0%	73.2%	18.2%	63.3%	79.7%	94.4%	83.1%	90.0%	63.0%	75.0%	65.8%
	MPQA	64.2%	21.4%	42.4%	48.0%	88.1%	11.2%	55.3%	54.0%	58.6%	83.0%	90.0%	45.0%	19.0%	52.3%
	Subj	50.8%	23.6%	71.2%	73.0%	74.6%	21.1%	71.2%	91.4%	95.6%	92.5%	90.0%	73.0%	51.0%	67.6%
	SST2	48.4%	50.4%	83.8%	68.0%	98.3%	1.9%	5.8%	99.2%	98.8%	99.6%	90.0%	64.0%	55.0%	66.4%
	yelp_reviews	58.6%	18.7%	76.2%	69.0%	80.3%	11.1%	65.3%	93.9%	95.8%	95.0%	90.0%	77.0%	60.0%	68.5%
	vader_nyt	39.8%	39.4%	72.4%	60.0%	69.2%	24.8%	64.6%	80.8%	90.9%	84.0%	90.0%	56.0%	58.0%	63.8%
	Average	44.6%	37.8%	76.3%	48.7%	81.4%	15.6%	56.2%	87.6%	90.9%	91.1%	90.0%	57.5%	60.2%	

Table 3.6: Percentage of reduction of the training set size.

For both domains (topics and sentiment), we notice that the methods XLDIS, CDIS, PSDSP, and LDIS have the highest reduction rates: on average 91.1%, 90.9%, 90.0% and 87.6%, respectively. The highest reduction rate is for XLDIS applied to SST1 (99.8%), while LSBo obtained the lowest reduction rate in this dataset. Thus, considering only the reduction criterion, these four algorithms stand out.

However, as we shall see, their negative impact on effectiveness is significant. According to the green scale, the lowest reduction rates are obtained by LSSM (on average 15.6%) followed by ENN (36.8%) and CNN (44.6%). In terms of datasets, the lowest reduction rates were obtained by LSSm on 20NG (0.5%), followed by the IB3 on the Reuters90 (1.0%) and 20NG (5.0%) and the LSBo applied on the SST2 (5.8%) and SST1 (7.7%). In addition, note that considering the global reduction rate (64.5%), the LSSm

approach has no above-average reduction rate in any dataset.

Based on these results, it is noticeable that all methods can reduce the training set to some extent – some more, others less. However, manipulations of the training set may have some profound impact (positive or negative) on the effectiveness of the classification models [31, 35, 37]. Table 3.7 answers the question *what is the impact on the reduction on effectiveness?* for each of the IS considered methods.

Considering the application of the IS methods to the best classification approach in each dataset (see Table 3.5), Table 3.7 presents the impact on MacroF1 regarding three situations: (1) the application of the IS methods produce statistically equivalent results to the classifier trained without any selection (NoSel) –represented in **bold** and with a green background; (2) the IS methods produce effectiveness losses that are under 5% when compared to NoSel – represented with an orange background. (3) the application of the IS method produced losses higher than 5% in effectiveness – represented in red.

In Table 3.7, instead of considering a simple statistical ("tie (win) vs. loss") binary scenario, we choose to include a third scenario for analysis, which includes an "acceptable loss". This acceptable loss corresponds to a scenario in which a potential reduction in training set size would compensate for the loss in effectiveness. For the sake of simplicity, here we considered a general, arbitrary rate of 5% of loss, which could be different for each scenario and situation. This 5% rate serves well our analysis purposes. As far as we know, there is no scientific methodology to calculate an acceptable loss. Indeed this may be completely dependent on the task and application. Thus, our choice for the 5% thresholds was based on a general assumption that, in practice, a loss of more than 5% would certainly hurt effectiveness, but a loss smaller than 5% could be acceptable in practice, given the benefits obtained from the use of IS methods in terms of reduction and efficiency. We leave for future work to determine a more suitable way to define a dataset-specific acceptable loss rate. Results with an orange background in Table 3.7 correspond to this scenario of "acceptable" loss at most 5%.

We start our analysis of results, shown in Table 3.7, by noting that none of the IS methods produced effectiveness improvements. In fact, NoSel was always the method with the highest absolute MacroF1 value on all datasets. This clearly shows that the tested IS methods did not reduce noise for the ATC task, at least in the tested textual datasets. In fact, despite the potential for noise removal motivation, selection methods were not able to improve the effectiveness of the text classification models. Even though these datasets being manually annotated, and they are traditional benchmarks scrutinized by the research community over the years, this does not mean that they are noise-free because human annotation is error-prone.

	dataset	NoSel	CNN	ENN	ICF	IB3	Drop3	LSSm	LSBo	LDIS	CDIS	XLDIS	PSDSP	EGDIS	CIS
Topic	**DBLP**	**81.7(0.5)**	79.1(0.8)	80.5(0.6)	77.6(0.8)	79.5(0.5)	78.4(0.6)	**81.1(0.8)**	79.1(0.6)	75.1(0.7)	75.6(0.6)	74.8(0.9)	60.6(0.5)	76.6(0.8)	74.0(1.3)
	Books	**89.5(0.2)**	85.9(1.5)	86.1(0.4)	84.1(0.5)	72.4(0.4)	84.3(0.5)	**88.8(0.5)**	84.0(0.5)	73.9(1.9)	72.8(2.3)	72.4(2.3)	79.7(0.5)	84.1(0.6)	80.3(0.5)
	ACM	**71.8(1.0)**	67.3(0.8)	67.2(1.9)	64.3(1.5)	66.6(0.6)	64.2(2.0)	**69.6(1.3)**	63.8(1.5)	62.8(1.3)	61.4(1.9)	60.6(2.2)	57.6(1.1)	65.7(1.1)	68.5(1.0)
	20NG	**90.7(0.6)**	**87.7(1.5)**	68.3(0.7)	68.9(1.0)	85.8(0.8)	79.2(2.3)	**90.7(0.5)**	**90.7(0.6)**	59.2(1.2)	60.7(1.3)	58.7(1.4)	89.0(0.6)	89.1(0.6)	89.5(0.7)
	OHSUMED	**77.8(1.2)**	73.3(0.4)	72.3(0.8)	67.8(0.9)	71.2(2.0)	68.3(1.2)	73.8(0.5)	68.8(1.2)	61.9(1.5)	62.4(1.3)	60.5(1.1)	58.5(1.3)	67.6(3.3)	61.2(2.0)
	Reuters90	**48.4(2.6)**	**46.9(2.5)**	33.1(2.1)	32.1(2.1)	**48.1(2.4)**	34.8(2.7)	38.1(1.7)	36.5(2.2)	33.5(1.4)	34.4(1.4)	33.1(1.2)	34.7(2.3)	45.3(2.5)	22.5(6.8)
	WOS-11967	**87.0(0.7)**	**85.0(1.2)**	85.1(0.5)	81.8(2.1)	84.7(0.8)	83.5(0.8)	**86.4(0.9)**	84.9(0.6)	79.1(0.8)	80.5(0.9)	77.2(1.9)	81.0(1.1)	84.3(0.9)	61.2(2.0)
	WebKB	**83.2(2.1)**	**81.9(1.6)**	76.7(1.8)	73.2(2.3)	**80.8(1.8)**	74.3(1.3)	**80.6(1.8)**	76.2(2.1)	61.6(2.8)	68.6(2.1)	60.1(2.9)	68.9(2.3)	**80.5(1.4)**	**80.5(1.9)**
	TREC	**95.5(0.5)**	**94.0(1.0)**	89.2(1.1)	88.9(1.5)	**93.8(1.3)**	87.2(2.5)	**95.0(0.7)**	**95.0(1.1)**	85.3(1.3)	87.2(1.7)	77.8(2.7)	88.8(1.4)	**92.5(3.2)**	92.4(0.4)
	WOS-5736	**90.5(0.9)**	**89.2(0.7)**	87.4(1.0)	84.9(0.9)	**88.4(1.0)**	85.5(1.1)	**88.0(1.1)**	86.5(1.4)	81.8(0.8)	83.7(0.8)	81.3(1.7)	82.2(0.8)	**88.4(1.3)**	55.4(9.9)
Sentiment	**SST1**	**53.8(1.3)**	48.0(1.4)	15.9(1.1)	21.0(1.2)	**53.3(1.0)**	32.0(1.4)	**53.4(0.9)**	**53.2(0.9)**	13.7(4.4)	26.1(4.3)	18.9(1.4)	48.9(1.1)	**53.4(1.0)**	**52.2(0.9)**
	pang_movie	**89.0(0.4)**	**88.2(0.8)**	**88.6(0.4)**	**87.6(0.6)**	87.1(0.6)	87.6(0.4)	**88.5(0.5)**	**88.0(0.6)**	86.9(0.7)	84.9(1.1)	86.6(0.7)	85.6(0.9)	86.8(0.8)	86.9(0.5)
	MR	**89.0(0.7)**	**63.6(15.4)**	37.3(1.9)	45.2(3.6)	87.3(0.8)	33.6(0.3)	**89.0(0.6)**	39.3(12.3)	40.6(9.9)	48.5(13.7)	33.9(1.1)	86.4(0.4)	86.5(1.0)	**88.0(0.6)**
	vader_movie	**91.3(0.5)**	**90.9(0.5)**	**90.6(0.5)**	**89.9(0.6)**	**91.3(0.7)**	**89.8(0.8)**	**90.8(0.7)**	**90.5(0.4)**	89.2(0.6)	**76.3(16.1)**	89.0(0.8)	88.4(0.6)	**89.9(0.6)**	89.1(0.8)
	MPQA	**90.2(0.8)**	**87.0(1.8)**	84.8(0.9)	85.9(0.5)	**88.7(0.7)**	86.6(1.6)	**90.0(0.7)**	**89.9(0.6)**	**89.1(0.6)**	**89.0(0.8)**	88.0(0.5)	**88.6(0.8)**	87.9(0.6)	**90.0(0.7)**
	Subj	**97.0(0.3)**	**96.4(0.5)**	95.9(0.4)	95.1(0.5)	95.7(0.6)	95.1(0.4)	95.4(0.7)	95.6(0.5)	41.9(13.7)	58.1(12.3)	44.5(14.2)	95.1(0.2)	96.2(0.4)	**96.7(0.4)**
	SST2	**93.2(0.6)**	60.7(11.7)	49.0(3.0)	61.0(4.7)	92.0(0.8)	39.6(5.0)	**92.9(0.5)**	**93.0(0.7)**	**60.4(17.2)**	**72.6(15.2)**	53.9(13.5)	90.2(1.0)	91.7(0.7)	92.0(0.8)
	yelp_reviews	**97.9(0.4)**	**97.2(0.3)**	**97.2(0.5)**	**97.2(0.4)**	**97.0(0.5)**	**96.9(0.4)**	**97.7(0.3)**	**97.4(0.3)**	94.7(1.1)	95.8(0.6)	94.7(1.0)	**96.8(0.6)**	**96.8(0.9)**	**97.3(0.4)**
	vader_nyt	**85.5(0.8)**	**83.8(1.1)**	83.4(1.3)	**82.8(1.3)**	83.6(0.7)	82.6(0.9)	**83.9(0.9)**	**83.6(1.2)**	**83.2(1.4)**	81.5(1.1)	82.2(1.1)	81.0(1.0)	83.2(1.0)	**84.0(0.9)**

Table 3.7: MacroF1 results. We present, for each dataset (row), the MacroF1 results of the application of IS approaches (columns) considering the best classification method for each dataset (Table 3.5). Cells with value in **bold** and with green background are statistically equivalent to the classification method without instance selection - **NoSel**. Furthermore, for each dataset, we present in the cells with orange background color the results with effectiveness up to 5% worse than the method without selection (NoSel) respectively.

Indeed, previous work has investigated the reasons for errors in automatic classification and found that noise is one of them [94]. For example, the **ACM** dataset comprises articles associated with the classification in the ACM taxonomy, as indicated by the author himself. Therefore, it is susceptible to the inherent noise of human labeling. Note that, despite the author being the person who knows the most about the proposed work, he may not know well the ACM taxonomy, which is complex and large [114]. Therefore, the taxonomy indicated by the author does not necessarily reflect the best class to assign to a document.

Another example is the Movie Review (MR) dataset. This dataset comprises movie reviews with users' associated sentiments on online platforms. Note that this context has considerable subjectivity, as it is based on users' tastes and preferences. In [88], the authors show, through a thorough analysis using Topic Modeling and Sentiment Analysis techniques, that the sentiment indicated in terms of the number of stars associated with movie reviews can often be noisy and not reflect the content of the comment made by the user. In addition, one may usually find negative and positive aspects of the movie in the same assessment, which confuses the classifier.

We will further investigate this issue in the future by analyzing the level of noise present in each dataset and how this relates to the current results.

We also observe in Table 3.7 that LSSm is the method that has more statistical ties – 16 times (different datasets) – compared to the classification using the complete training set. Evaluating the results on the topic datasets, LSSm can maintain effectiveness in 8 out of 10 cases. The result is even better when we consider the sentiment datasets, on which LSSm is equivalent to NoSel in 8 out of 9 datasets. The CNN method can also obtain statistically equivalent results in 13 of the 19 datasets. More specifically, CNN achieves equivalent results in more than half of the topics and on 7 out of 9 sentiment datasets. LSBo in turn obtains statistically equivalent results in 9 of the 19 datasets – it achieves statistically equivalent results in 2 topic datasets and, like CNN, in 7 of 9 sentiment datasets. Following, the IB3 method can obtain statistically equivalent results in 8 of the 19 datasets (four ties in both, sentiment and topic tasks). CIS method in turn obtains statistically equivalent results in 7 of the 19 datasets, however, it achieves statistically equivalent results in only one topic and 6 of 9 sentiment datasets.

For both topic and sentiment domains, XLDIS, PSDSP, LDIS, ICF, and CDIS did not perform well, being only able to tie with NoSel in a maximum of 4 different datasets. More specifically, when considering topic classification, none of those approaches could achieve NoSel results. Among the above methods, XLDIS can be considered the worst in terms of effectiveness, as this approach does not achieve equivalent results in any dataset. In general, we have between 0 to 10 methods (out of 13) in each dataset that are statistically equivalent to NoSel - OHSUMED and yelp_reviews, respectively.

Let´s now consider scenario 2 – the IS methods produce effectiveness losses under

5% compared to NoSel — represented with an orange background. We can note that LSSm remains the best method for maintaining effectiveness, being equivalent in 17 of the 19 datasets. Only in two datasets (Reuters90 and OHSUMED - topic classification task), LSSm had significant losses. Moreover, considering sentiment, LSSM achieves excellent results – equivalent in all 9 datasets. The CNN and LSBo methods follow, respectively, with 15 and 13 equivalent datasets. CNN achieves equivalent results in 8 out of 10 topic datasets and maintains the results in 7 out of 9 sentiment datasets. LSBo, in turn, achieves equivalent results in 5 topic datasets and 8 out of 9 sentiment datasets. Following, the IB3 and EGDIS methods can obtain equivalent results in, respectively, 15 and 14 of the 19 datasets. IB3 achieves equivalent results in 6 out of 10 topic datasets and maintains the results in all sentiment datasets. On the other hand, EGDIS achieves equivalent results in just one dataset less than IB3 considering the topic results and, also, maintains the results in all sentiment datasets.

When considering this "acceptable" loss of 5%, the XLDIS, PSDSP, LDIS, ICF, and CDIS have a significant improvement, reaching 30 ties altogether – previously, there were only 12. Note however that these results are achieved mostly for the sentiment datasets. Finally, we note that XLDIS maintains its behavior, being the worst method regarding effectiveness (only 5 ties). Besides, note that when we consider the LDIS, CDIS, XLDIS, and PSDSP, we must always be careful to consider an eventual loss of 5% or more, since, in 16 out of 36 outputs, there were losses in terms of effectiveness or some instability for these methods, captured by a high confidence interval.

Besides, these four methods are based on measuring density and/or partitions in the hyperplane of the feature space. In this case, the more separable the elements that compose the classes of a dataset, the more effective the selection of instances that will be considered. Therefore, our main hypothesis is that due to the tabular nature of these datasets (in terms of both the number of instances and features), they can be considered more separable problems than the ones usually found in the context of ATC.

To corroborate our hypothesis, we mention an analysis [17] in which the authors demonstrate that these IS algorithms achieved satisfactory results when applied to less complex tasks (i.e., structured data). ATC, however, deals with more complex datasets with high sparsity and high dimensionality. One commonly mentioned issue is the "Curse of dimensionality", where the space is so ample that the instances are equivalently far away from each other. This type of phenomenon naturally disturbs the concepts of density and the feature hyperplane partition.

The experimental results tend to corroborate our hypothesis. We can see some cases (8 out of 36) where these methods perform well (e.g. yelp_reviews), especially when considering the sentiment analysis task. Considering the MPQA dataset, for instance, three of the four approaches achieved statistically equivalent results. Note that this dataset has the lowest dimensionality among those tested in our work. As a sec-

ond example, Yelp_reviews has a relatively high density (average number of terms per document), which directly benefited the result achieved by PSDSP (based on hyperplane partitions).In general, when we consider topic classification, regardless of the specific application, it is possible to observe that only CNN and LSSm satisfy the effectiveness requirement. On the other hand, for the sentiment domain, we can stratify this result. For user reviews, movie and product reviews (pang_movie, vader_movie, yelp_reviews), if a 5% loss is allowed, all algorithms perform well.

task	dataset	NoSel	CNN	ENN	ICF	IB3	Drop3	LSSm	LSBo	LDIS	CDIS	XLDIS	PSDSP	EGDIS	CIS
Topic	**DBLP**	1.5	3.5	3.5	7.5	5.5	7.5	1.5	5.5	11.0	11.0	11.0	14.0	11.0	11.0
	Books	1.5	3.5	3.5	7.5	12.5	5.5	1.5	5.5	12.5	12.5	12.5	9.5	7.5	9.5
	ACM	1.5	5.0	5.0	8.5	8.5	11.5	1.5	5.0	11.5	11.5	11.5	14.0	5.0	5.0
	20NG	2.5	2.5	10.5	10.5	8.0	9.0	2.5	2.5	13.0	13.0	13.0	6.0	6.0	6.0
	OHSUMED	1.0	3.5	3.5	8.0	6.5	8.0	3.5	3.5	11.5	11.5	11.5	14.0	6.5	11.5
	Reuters90	2.0	2.0	11.5	11.5	2.0	6.5	6.5	6.5	11.5	11.5	11.5	6.5	4.0	11.5
	WOS-11967	2.0	2.0	6.0	11.0	6.0	6.0	2.0	6.0	11.0	11.0	11.0	11.0	6.0	14.0
	WebKB	3.5	3.5	8.0	8.0	3.5	10.0	3.5	8.0	12.0	12.0	14.0	12.0	3.5	3.5
	TREC	3.5	3.5	11.5	8.0	3.5	8.0	3.5	3.5	11.5	11.5	14.0	11.5	3.5	8.0
	WOS-5736	3.0	3.0	7.0	7.0	3.0	10.0	3.0	7.0	13.0	10.0	10.0	13.0	3.0	13.0
	Average	2.2	3.2	7.0	8.8	5.9	8.2	2.9	5.3	11.9	11.6	12.0	11.2	5.6	9.3
Sentiment	**SST1**	3.5	7.5	14.0	12.0	3.5	9.5	3.5	3.5	12.0	9.5	12.0	7.5	3.5	3.5
	pang_movie	3.5	3.5	3.5	3.5	10.5	10.5	3.5	3.5	10.5	10.5	10.5	10.5	10.5	10.5
	MR	2.5	2.5	12.5	9.0	5.0	12.5	2.5	9.0	12.5	9.0	12.5	5.0	5.0	2.5
	vader_movie	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	12.5	5.5	12.5	12.5	5.5	12.5
	MPQA	5.0	5.0	12.0	12.0	5.0	12.0	5.0	5.0	5.0	5.0	12.0	5.0	12.0	5.0
	Subj	2.0	2.0	7.0	7.0	7.0	7.0	7.0	7.0	13.0	13.0	13.0	11.0	7.0	2.0
	SST2	3.0	10.5	13.0	10.5	7.5	13.0	3.0	3.0	3.0	3.0	13.0	7.5	7.5	7.5
	yelp_reviews	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	11.0	11.0	11.0	6.0	6.0	6.0
	vader_nyt	4.0	4.0	11.0	4.0	11.0	11.0	4.0	4.0	4.0	11.0	11.0	11.0	11.0	4.0
	Average	3.9	5.2	9.4	7.7	6.8	9.7	4.4	5.2	9.3	8.6	11.9	8.4	7.6	5.9
Overall	**Aggr. Ranking**	3.0	4.1	8.1	8.3	6.3	8.9	3.6	5.2	10.6	10.2	12.0	9.9	6.5	7.7

Table 3.8: Instance Selection MacroF1 - Fractorial Ranking Results.

To better summarize the effectiveness results, we rank the best IS techniques by ordering each IS approach in each dataset considering the MacroF1 metric. The Fractional Ranking method was used to compare multiple methods applied to each dataset. In practice, the method works as follows: for each fold of each dataset, the best method is ranked; the best - rank in rank 1, the second-best - rank 2, and so on. In case of a statistical tie between methods, we assign the average of their respective rankings to each method. For instance, two methods tied in the first place get both rank 1.5. The result is shown in the table 3.8. As a "rule of thumb", the best-ranked method in each dataset should be NoSel, along with potential ties.

As expected, in Table 3.8 the NoSel aggregated ranking is the overall best – 3.0 on average for all datasets (top-ranking). We note that few IS strategies approximate the NoSel results. Overall, the three strategies that come closest to NoSel are LSSm (Aggr. Ranking - 3.6), CNN (4.1), and LSBo (5.2). In fact, in several datasets, these IS strategies achieve results very close to NoSel, which is evidenced by the several ranking ties in the

first place. For instance, in 20NG and TREC, the three best selection methods (according to the Aggr. Ranking) are tied with NoSel.

Briefly summarizing, both experiments indicate an affirmative answer for RQ1 – there are selection methods capable of reducing the training set while maintaining effectiveness in general.

3.3.2 RQ1.2. What is the impact of applying IS strategies on the text classification models' total construction time?

Intuitively, there is an expectation that the selection of the most representative instances of the training set will lead to a reduction in model construction time. We verified, by answering RQ1, that the IS methods were not able to improve the effectiveness of the models. However, there is a subset of IS methods with the potential to reduce the training set while **maintaining** the effectiveness (RQ1), more specifically: LSSm, CNN, and LSBO. However, introducing an IS step in the pre-construction phase of the model may be costly, thus causing overhead in terms of time. Using the IS method can be even more expensive than building the model with all the data if the IS step is not cheap enough. Note that each strategy impacts application time, as we consider the total cost – preprocessing + IS application + training time to build the model. Therefore, for the use of IS methods to be attractive, they must provide, at a minimum, efficiency improvements. Thus, in the subsequent analysis, we assess how much it costs to reduce the total time of the training set in terms of model construction. In Table 3.9, we show, for each dataset, the total application speedup (NoSel Time / IS approach Time) of each IS approach with its respective (best) classifier (see Table 3.5). For each dataset (line), we have a color scale, where the greener, the higher speedup, and the redder, the higher the computational cost (average execution time) compared to NoSel.

The algorithms on the right side of the table – LDIS, CDIS, XLDIS, and PSDSP – are the most efficient in terms of speedup. They were also the ones that reduced the training set the most. However, despite the attractive cost and reduction rate, these algorithms were, considering an acceptable loss of 5%, able to maintain effectiveness in a maximum of 31.5% of cases (24 of 76 cells). Considering only the statistically equivalent results to NoSel, the overall result is even worse – 11.8% (9 of 76 cells).

Next, we concentrate our efficiency analysis on the best IS approaches according to the fractional ranking presented for (RQ1): LSSm, CNN, and LSBO. As seen in Table 3.7, LSSm achieved good results in terms of effectiveness, but it could not produce large training set reductions. Consequently, as we can visually grasp, LSSm is the most

task	dataset	CNN	ENN	ICF	IB3	Drop3	LSSm	LSBo	LDIS	CDIS	XLDIS	PSDSP	EGDIS	CIS
Topic	DBLP	1.10	0.92	0.99	0.68	1.43	0.83	1.11	4.91	6.11	5.69	5.45	1.83	0.10
	Books	1.04	0.83	0.80	0.61	1.33	0.80	1.09	5.02	5.54	5.90	4.70	1.91	0.25
	ACM	1.44	1.11	1.44	1.12	1.98	0.94	1.35	4.07	5.04	5.82	7.02	1.94	0.46
	20NG	0.94	1.91	1.67	0.52	2.29	0.78	0.72	91.87	37.90	92.61	51.73	2.20	0.67
	OHSUMED	1.49	1.03	1.84	1.38	2.16	1.06	1.89	6.97	7.19	6.74	7.27	1.58	0.39
	Reuters90	2.10	3.55	8.33	0.88	10.35	2.11	5.69	45.03	32.05	62.05	40.52	2.60	1.56
	WOS-11967	1.38	1.08	2.30	1.56	2.52	1.06	2.20	7.50	7.15	7.99	5.76	2.11	0.87
	WebKB	1.39	1.10	1.98	1.37	2.41	1.09	2.36	5.88	6.35	6.34	6.25	1.63	0.75
	TREC	1.30	1.79	2.24	1.23	3.67	1.12	1.24	5.65	4.70	14.09	6.88	1.31	0.21
	WOS-5736	1.54	1.02	2.66	1.78	2.81	1.09	2.30	6.15	5.66	5.68	5.27	2.08	1.33
Sentiment	SST1	1.22	2.86	5.23	0.89	12.59	0.95	0.78	22.68	15.60	24.76	5.96	1.21	0.21
	pang_movie	1.49	1.32	1.70	1.55	2.41	1.05	1.57	3.72	5.89	4.01	4.58	2.13	0.53
	MR	1.19	1.07	2.30	1.53	10.10	0.92	1.09	14.04	10.72	14.71	5.11	2.03	0.28
	vader_movie	1.59	1.21	1.68	0.89	2.33	1.09	1.54	3.60	6.70	4.14	4.59	2.12	0.53
	MPQA	2.18	0.84	0.71	0.85	3.35	0.86	1.33	1.83	1.73	3.64	6.21	1.60	0.07
	Subj	1.63	1.18	1.75	1.87	2.42	1.07	1.72	3.62	5.50	4.25	5.47	2.90	0.52
	SST2	1.46	1.29	2.44	1.80	10.86	0.87	0.81	14.32	10.08	16.36	5.73	2.21	0.31
	yelp_reviews	2.09	0.97	2.69	2.84	3.15	1.15	2.30	6.79	7.83	7.16	5.00	3.13	1.45
	vader_nyt	1.50	1.42	2.30	1.55	2.33	1.19	2.09	4.01	5.77	4.40	5.29	1.62	0.98
	Average	1.48	1.40	2.37	1.31	4.24	1.05	1.75	13.56	9.87	15.60	9.94	2.01	0.60

Table 3.9: SpeedUp on Total Application Cost of the Instance Selection Methods applied to the best ATC approach in each dataset.

costly method (predominantly light green with several red cells). Its low reduction rate, added to its high computational cost, makes the process as a whole unfeasible. There are no gains in effectiveness and its application in the pipeline worsens the overall execution time. The average speed-up for this approach is **1.05** (varying between **0.78** and **2.11**). Even though LSSm was able to produce equivalent results to NoSel in 16 out of 19 datasets (Table 3.7), according to Table 3.9, it produced only 11-time improvements. If we consider only the results in which LSSm statistically ties with NoSel, there are time improvements in only 8 datasets. If we consider an acceptable loss of 5% in effectiveness, nine datasets have a time improvement (one more). In sum, in roughly half of the cases, the incorporation of LSSm into the process takes longer than NoSel.

In the case of LSBo, although it has nine statistical ties in effectiveness with NoSel (7 less than LSSm), this method has an average speedup of **1.75** – higher than LSSm's. The method meets both requirements (effectiveness and efficiency) in six datasets but worsens the time in 3 others (20NG, SST1, SST2), being better than LSSm in six datasets (TREC, pang_movie, vader_movie, yelp_reviews, vader_nyt, and MPQA). If we consider an acceptable loss of 5% in effectiveness, the method can fulfill both constraints in 10 datasets. In this sense, based on the number of times LSBo and LSSm meet all requirements, LSBO can be considered better than LSSm.

The traditional CNN method was able to achieve total time improvements in **12** of the 13 datasets where it is also statistically equivalent to **NoSel** regarding effectiveness – the only case that does not improve in time is 20NG. As seen earlier in Table 3.6, CNN has an average reduction rate of **44.6%**, which positively impacts in terms of time. This method has an average speedup of **1.48**, better than LSSm in this regard and slightly

worse than LSBo. If we consider an acceptable loss of 5% in effectiveness (achieved by this method in 15 out of 19 datasets), CNN can improve time in **14** of these 15 datasets, a very good result.

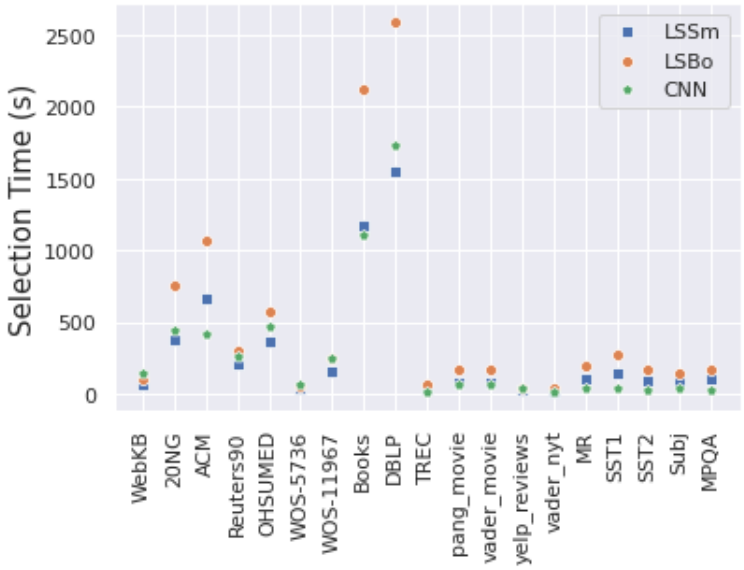


Figure 3.2: CNN, LSBo, and LSSm selection time (in seconds)

As a final analysis, we present in Figure 3.2 the selection generation step time (in seconds) for LSSm, LSBo, and CNN in each dataset. We note that, regardless of the domain (topic or sentiment), the most expensive method, i.e., the one that generally takes the longest to generate the selection set, is LSBo. LSSm assumes the middle position between LSBo and CNN for sentiment datasets and is, in general, the fastest method in the topic datasets. Finally, CNN is usually at the bottom edge of the figure, mainly for sentiment datasets while in the topic datasets it is mostly the runner-up.

In sum, given all analyzed results and overall tradeoff analyses, we consider **CNN** the best IS method evaluated in this Chapter. CNN can maintain effectiveness (or produce minimal losses), reduce the training set, and improve efficiency in several datasets. Note, however, that this result is still far from ideal, as CNN can fulfill all requirements in only 63% of the cases. This leaves plenty of room for the development of new IS methods specially designed for meeting all requirements in the context of ATC.

Additional Observations. In terms of effectiveness (MacroF1), we ranked the IS methods considering a general result (Aggr. Ranking) and also separating the topic and sentiment domains using Fractional Ranking methods (Table 3.8). Therefore, regardless of the sub-task, we observe the superiority of CNN, LSBo, and LSSm for sentiment analysis.

3.3.3 RQ1.3. How do IS approaches behave when applied to neural classification methods (especially Transformers)?

While answering the preliminary question (Section 3.2), we observed that deep learning classification methods reached the best results in 17 of the 19 considered datasets. Next, we investigate an issue related to the impact of fine-tuning on these methods since training data is mostly used in such architectures to fine-tune a general, pre-trained model to the characteristics of a given dataset/task.

3.3.3.1 Is the Fine-Tuning step really necessary for Automatic Text Classification?

IS methods reduce time because they reduce the number of documents for training. Accordingly, training set reduction can potentially reduce the time of the fine-tuning step (aka knowledge transfer), the most costly step for most deep learning methods. On the other hand, there is a growing area called Zero-Shot [13] learning, where deep learning approaches are used in an unsupervised way (without the use of fine-tuning) for a possible association of document embeddings to semantic classes. Therefore, the question naturally arises: "What is the impact of fine-tuning? Is it really necessary for achieving high effectiveness?".

There have been reports in several application fields [28, 140] in which Zero-Shot is useful and does not imply on effectiveness losses. On the other hand, while conducting our comparative investigation we observed that Zero-Shot learning is not very effective for the ATC. Table 3.10 has resulted from the application of a Zero-Shot learning[7] using RoBERTa's with the previously used datasets.

Note that the application of the RoBERTa model with fine-tuning has a significant impact on effectiveness. Compared to the ZeroShot model, fine-tuning leads to average gains of 900%, ranging from 121% to 3141% in MacroF1. Therefore, fine-tuning is essential in terms of effectiveness. On the other hand, the ZeroShot model has a significant impact on the model application time (Table 3.11), as there is an additional domain transfer step. The fine-tuned model can take between 4.5x to 105x longer when compared to the ZeroShot model. In short, it is clear that, although costly, fine-tuning is extremely important to the context of ATC. Therefore, IS methods can potentially benefit a lot from the reduc-

[7] Code: https://shorturl.at/aipu3

Dataset	ZeroShot RoBERTa	RoBERTa **with** finetunning	MacroF1 Gains(%)	Dataset	ZeroShot RoBERTa	RoBERTa **with** finetunning	MacroF1 Gains(%)
WebKB	10.91(0.9)	83.0(2.0)	660.77%	**pang_movie**	33.39(0.1)	89.0(0.4)	166.55%
20NG	3.46(0.1)	86.8(0.7)	2408.67%	**vader_movie**	33.48(0.3)	91.3(0.5)	172.70%
ACM	11.90(0.1)	70.3(1.4)	490.76%	**yelp_reviews**	33.54(0.4)	97.9(0.4)	191.89%
Reuters90	3.5(0.2)	41.9(2.2)	1097.14%	**vader_nyt**	27.84(0.3)	85.3(0.6)	206.39%
OHSUMED	2.4(0.4)	77.8(1.2)	3141.67%	**MR**	33.67(0.2)	89.0(0.7)	164.33%
WOS-5736	3.11(0.7)	90.5(0.9)	2809.97%	**SST1**	4.51(0.2)	53.8(1.3)	1092.90%
WOS-11967	11.47(1.1)	86.8(0.4)	656.76%	**SST2**	35.46(0.2)	93.2(0.6)	162.83%
Books	8.22(0.5)	87.2(0.6)	960.83%	**Subj**	43.81(1.3)	96.9(0.4)	121.18%
DBLP	6.63(0.4)	81.4(0.5)	1127.75%	**MPQA**	15.0(0.1)	90.2(0.8)	501.33%
TREC	8.89(10.7)	95.5(0.5)	974.24%				

Table 3.10: ZeroShot Analysis – MacroF1 Metric

Dataset	ZeroShot RoBERTa	RoBERTa **with** finetunning	Time Inc.	Dataset	ZeroShot RoBERTa	RoBERTa **with** finetunning	Time Inc.
WebKB	32.6	602.5	18.5x	**pang_movie**	13.9	681.3	49.2x
20NG	253.3	2781.8	11.0x	**vader_movie**	13.2	675.4	51.4x
ACM	67.9	3050.3	44.9x	**yelp_reviews**	7.2	760.7	105.2x
Reuters90	477.6	2156.9	4.5x	**vader_nyt**	6.2	332.0	53.2x
OHSUMED	301.0	2780.1	9.2x	**MR**	12.4	672.4	54.1x
WOS-5736	48.7	820.2	16.8x	**SST1**	18.2	809.6	44.4x
WOS-11967	273.8	1759.6	6.4x	**SST2**	12.2	619.9	50.7x
Books	242.8	4412.5	18.2x	**Subj**	12.8	708.5	55.4x
DBLP	187.6	4988.1	26.6x	**MPQA**	12.9	676.6	52.6x
TREC	9.4	463.5	49.3x				

Table 3.11: ZeroShot Analysis – Time (seconds) and SpeedUp

tion that IS methods may induce in the training sets used to fine-tune the NN methods.

As a result, we demonstrate fine-tuning classification is essential to achieve better results. One of the motivations for our work was that several authors claim that deep learning models need large amounts of data in the fine-tuning stage to properly learn [103, 76, 46]. As IS methods have shown to be beneficial in several situations, we want to further investigate this issue, focusing on Transformer Architectures that have produced top-notch ATC results in the literature.

3.3.3.2 Does the Fine-tuning phase of DL models need a lot of data as generally accredited in the literature or is a "right and carefully selected" training set enough for producing high effectiveness?

We have found at least one work that shows that it is possible to fine-tune a large NN model with fewer data in the image domain. In [27] the authors introduce the Sim-

CLRv2 method that performs, as an intermediate step, a fine-tuning with a small fraction of data that has class labels (few labeled examples). However, the image domain differs from the textual one in important ways, including the very high dimensionality of textual vocabularies, the variability in the number of features that a set of documents may contain, and the impact that noisy and ambiguous words may have on effectiveness. As evidence of such differences, convolutional methods (CNNs) brought numerous advances to the state-of-the-art in the image domain, but the same advances were not observed in textual applications [149, 71]. However, the success of that particular work in reducing the training set for NN training motivates a deeper study on the potential impact of IS methods in the fine-tuning phase of modern Transformer architectures for ATC.

We have previously applied the IS methods to the best ATC algorithms for each specific dataset. That is, the focus was on the IS algorithms. In this Section, we invert the perspective and focus on Transformer architectures that present the overall best effectiveness across most datasets, applying the best IS methods (CNN, LSSm, and LSBo) to those Transformer architectures. As before, we analyze the results in terms of the previously described tripod constraints.

Task	dataset	TFIDF	MF+SVM	BERT	XLNet	RoBERTa	GPT2	DistilBert	AlBERT	BART
Topic	**DBLP**	7.5	6.0	3.0	3.0	3.0	7.5	3.0	9.0	3.0
	Books	9.0	2.0	1.0	4.5	4.5	7.5	4.5	7.5	4.5
	ACM	8.0	3.5	3.5	3.5	3.5	8.0	3.5	8.0	3.5
	20NG	2.5	1.0	7.0	2.5	5.0	8.0	5.0	9.0	5.0
	OHSUMED	8.5	8.5	3.0	3.0	3.0	6.5	3.0	6.5	3.0
	Reuters90	9.0	1.0	4.5	4.5	4.5	8.0	4.5	4.5	4.5
	WOS-11967	6.0	7.5	5.0	2.5	2.5	7.5	2.5	9.0	2.5
	WebKB	8.5	8.5	3.5	3.5	3.5	7.0	3.5	3.5	3.5
	TREC	8.5	8.5	7.0	3.0	3.0	6.0	3.0	3.0	3.0
	WOS-5736	3.5	7.5	3.5	3.5	3.5	9.0	3.5	7.5	3.5
Sentiment	**SST1**	8.5	8.5	2.5	2.5	2.5	7.0	5.5	5.5	2.5
	pang_movie	9.0	8.0	5.0	2.5	2.5	7.0	6.0	2.5	2.5
	MR	8.0	9.0	2.5	2.5	2.5	7.0	5.5	5.5	2.5
	vader_movie	8.0	7.0	4.0	2.0	2.0	6.0	6.0	6.0	2.0
	MPQA	9.0	8.0	2.0	5.0	2.0	7.0	5.0	5.0	2.0
	Subj	9.0	8.0	3.0	3.0	3.0	7.0	3.0	6.0	3.0
	SST2	8.5	8.5	4.0	4.0	1.5	7.0	6.0	4.0	1.5
	yelp_reviews	5.5	9.0	5.5	2.0	2.0	8.0	5.5	5.5	2.0
	vader_nyt	8.0	9.0	4.5	3.0	1.5	6.5	4.5	6.5	1.5
Overall	**Aggr. Ranking**	7.6	6.8	3.9	**3.2**	**2.9**	7.2	4.4	6.0	**2.9**

Table 3.12: Classification Approaches - MacroF1 Fractorial Ranking Results.

To determine the best overall Transformer methods across all datasets, we generate the MacroF1 Fractional Ranking analysis for the classification approaches. The results presented in Table 3.12 show that, in terms of effectiveness, the classification approaches that, on average, produce the best MacroF1 results are: **RoBERTa** (Aggr. Ranking 2.9), **BART** (2.9) and **XLNet** (3.2). Note that, when looking at results presented previously in Table 3.4, these three classifiers are exactly the ones that have the highest number of statistical ties when compared to the best approach in each dataset.

		RoBERTa									
		NoSel	CNN			LSSm			LSBo		
task	Dataset	MacF1	MacF1	Reduction	SpeedUp	MacF1	Reduction	SpeedUp	MacF1	Reduction	SpeedUp
Topic	DBLP	**81.4(0.5)** ▲	79.0(0.5)	52.4%	1.17x	80.8(0.7)	17.4%	0.87x	78.6(0.9)	72.8%	1.23x
	Books	**87.2(0.6)** •	83.4(1.7)	32.1%	1.05x	**86.5(0.6)** •	8.8%	0.88x	81.5(0.7)	63.7%	1.22x
	ACM	**70.3(1.4)** ▲	65.4(1.4)	47.1%	1.48x	68.0(1.3)	19.0%	1.00x	63.4(1.6)	67.7%	1.37x
	20NG	**86.8(0.7)** •	81.6(1.1)	27.9%	1.13x	**86.9(0.5)** •	0.5%	0.87x	85.6(0.6)	23.2%	1.00x
	OHSUMED	**77.8(1.2)** ▲	73.3(0.4)	45.5%	1.49x	73.8(0.5)	21.9%	1.06x	68.8(1.2)	69.8%	1.89x
	Reuters90	**41.9(2.2)** •	**41.7(3.1)** •	50.7%	1.46x	**41.3(2.1)** •	28.4%	1.37x	**41.3(2.1)** •	76.9%	1.30x
	WOS-11967	**86.8(0.4)** •	85.6(0.7)	45.4%	1.37x	**86.5(0.6)** •	22.1%	0.99x	85.2(0.8)	68.4%	1.89x
	WebKB	**83.0(2.0)** ▲	79.8(1.4)	42.9%	1.10x	78.9(2.1)	24.1%	1.02x	72.5(2.7)	71.1%	1.82x
	TREC	**95.5(0.5)** •	**94.0(1.0)** •	31.3%	1.30x	**95.0(0.7)** •	18.4%	1.12x	**95.0(1.1)** •	37.8%	1.24x
	WOS-5736	**90.5(0.9)** •	**89.2(0.7)** •	50.4%	1.54x	**88.0(1.1)** •	20.1%	1.09x	86.5(1.4)	70.9%	2.30x
Sentiment	SST1	**53.8(1.3)** •	48.0(1.4)	18.9%	1.22x	**53.4(0.9)** •	5.7%	0.95x	**53.2(0.9)** •	7.7%	0.84x
	pang_movie	**89.0(0.4)** •	**88.2(0.8)** •	46.8%	1.49x	**88.5(0.5)** •	18.8%	1.05x	**88.0(0.6)** •	63.5%	1.57x
	MR	**89.0(0.7)** •	**63.6(15.4)** •	46.7%	1.19x	**89.0(0.6)** •	3.3%	0.92x	39.3(12.3)	48.8%	1.09x
	vader_movie	**91.3(0.5)** •	**90.9(0.5)** •	47.2%	1.59x	**90.8(0.7)** •	18.1%	1.09x	**90.5(0.4)** •	63.3%	1.54x
	MPQA	**90.2(0.8)** •	**87.0(1.8)** •	64.2%	2.18x	**90.0(0.7)** •	11.2%	0.86x	**89.9(0.6)** •	55.3%	1.33x
	Subj	**96.9(0.4)** •	**96.1(0.8)** •	50.8%	1.79x	95.1(0.5)	21.1%	1.07x	95.3(0.4)	71.2%	1.98x
	SST2	**93.2(0.6)** •	60.7(11.7)	48.4%	1.46x	**92.9(0.5)** •	1.9%	0.87x	**93.0(0.7)** •	5.8%	0.81x
	yelp_reviews	**97.9(0.4)** •	**97.2(0.3)** •	58.6%	2.09x	**97.7(0.3)** •	11.1%	1.15x	**97.4(0.3)** •	65.3%	2.30x
	vader_nyt	**85.3(0.6)** •	**84.9(1.1)** •	39.8%	1.44x	**84.8(1.2)** •	24.9%	1.24x	83.6(1.3)	64.6%	1.86x

Table 3.13: Effectiveness, reduction and speedup Analysis. We present for each dataset (row) the MacroF1 results of the application of CNN, LSSm and LSBo IS approaches (columns) considering the **RoBERTa** classifier. Cells with value in **bold** and with a green background are statistically equivalent to the MacroF1 columns with the higher value (marked as •or ▲). Furthermore, for each dataset, we present in the cells with orange background color the results with effectiveness up to 5% worse than the higher MacF1 column, respectively.

		BART									
		NoSel	CNN			LSSm			LSBo		
task	Dataset	MacF1	MacF1	Reduction	SpeedUp	MacF1	Reduction	SpeedUp	MacF1	Reduction	SpeedUp
Topic	DBLP	**81.1(0.5)** ●	78.7(0.7)	52.4%	1.20x	**80.6(0.6)** ●	17.4%	0.96x	78.5(0.7)	72.8%	1.33x
	Books	**86.9(0.5)** ●	82.4(1.7)	32.1%	1.02x	**86.0(0.4)** ●	8.8%	0.78x	80.7(0.7)	63.7%	0.96x
	ACM	**70.8(0.7)** ▲	66.2(1.5)	47.1%	1.48x	68.7(1.5)	19.0%	0.96x	63.7(1.1)	67.7%	1.51x
	20NG	**87.4(0.9)** ●	82.2(1.5)	27.9%	1.25x	**87.1(0.9)** ●	0.5%	0.95x	86.3(0.9)	23.2%	1.08x
	OHSUMED	**77.6(0.7)** ▲	73.5(1.1)	45.5%	1.46x	74.9(0.8)	21.9%	1.10x	70.2(1.1)	69.8%	1.89x
	Reuters90	**42.2(2.1)** ●	**42.2(2.0)** ●	50.7%	1.47x	**42.3(2.5)** ●	28.4%	1.45x	**42.3(2.5)** ●	76.9%	1.38x
	WOS-11967	**86.9(0.8)** ●	85.6(0.6)	45.4%	1.31x	**86.5(0.9)** ●	22.1%	0.87x	84.9(0.6)	68.4%	1.96x
	WebKB	**83.0(1.7)** ▲	80.3(1.6)	42.9%	1.36x	80.1(1.8)	24.1%	1.17x	75.4(1.8)	71.1%	2.08x
	TREC	**95.5(0.8)** ●	93.6(1.3)	31.3%	1.41x	**94.5(1.1)** ●	18.4%	1.10x	**94.3(1.0)** ●	37.8%	1.33x
	WOS-5736	**89.6(1.7)** ●	**88.7(1.4)** ●	50.4%	1.69x	**87.6(1.0)** ●	20.1%	0.83x	85.8(1.2)	70.9%	2.56x
Sentiment	SST1	**52.8(1.0)** ●	46.2(1.5)	18.9%	1.13x	**52.6(1.0)** ●	5.7%	0.86x	**52.3(0.7)** ●	7.7%	0.80x
	pang_movie	**88.1(0.5)** ●	**87.3(0.6)** ●	46.8%	1.58x	**88.0(0.6)** ●	18.8%	1.06x	**87.3(0.6)** ●	63.5%	1.73x
	MR	**88.2(0.6)** ●	67.5(13.4)	46.7%	1.74x	**88.3(0.4)** ●	3.3%	0.96x	39.0(12.1)	48.8%	1.09x
	vader_movie	**90.4(0.6)** ●	**89.7(0.5)** ●	47.2%	1.57x	**89.8(0.6)** ●	18.1%	1.06x	89.1(0.4)	63.3%	1.62x
	MPQA	**90.1(0.7)** ●	87.5(1.5)	64.2%	2.53x	**90.2(0.7)** ●	11.2%	0.96x	**90.0(0.7)** ●	55.3%	1.56x
	Subj	**96.8(0.4)** ●	**96.3(0.4)** ●	50.8%	2.28x	95.6(0.5)	21.1%	1.47x	95.7(0.6)	71.2%	2.66x
	SST2	**92.8(0.5)** ●	71.1(5.2)	48.4%	1.69x	**92.4(0.5)** ●	1.9%	0.89x	**92.7(0.5)** ●	5.8%	0.86x
	yelp_reviews	**97.5(0.4)** ●	**97.3(0.3)** ●	58.6%	2.48x	**97.2(0.4)** ●	11.1%	1.25x	**97.5(0.3)** ●	65.3%	2.26x
	vader_nyt	**85.5(0.8)** ●	**83.8(1.1)** ●	39.8%	1.50x	**83.9(0.9)** ●	24.9%	1.19x	**83.6(1.2)** ●	64.6%	2.09x

Table 3.14: Effectiveness, reduction and speedup Analysis. We present for each dataset (row) the MacroF1 results of the application of CNN, LSSm and LSBo IS approaches (columns) considering the **BART** classifier. Cells with value in **bold** and with a green background are statistically equivalent to the MacroF1 columns with the higher value (marked as ●or ▲). Furthermore, for each dataset, we present in the cells with orange background color the results with effectiveness up to 5% worse than the higher MacF1 column, respectively.

		XLNet									
		NoSel	CNN			LSSm			LSBo		
task	Dataset	MacF1	MacF1	Reduction	SpeedUp	MacF1	Reduction	SpeedUp	MacF1	Reduction	SpeedUp
Topic	DBLP	**81.4(0.6)** ●	78.9(0.6)	52.4%	1.47x	**81.1(0.6)** ●	17.4%	1.09x	79.1(0.6)	72.8%	1.73x
	Books	**87.3(0.4)** ●	83.4(1.4)	32.1%	1.15x	**87.0(0.5)** ●	8.8%	0.88x	80.8(0.6)	63.7%	1.44x
	ACM	**69.9(0.9)** ▲	64.2(2.1)	47.1%	1.65x	67.5(1.5)	19.0%	0.97x	62.2(1.1)	67.7%	1.82x
	20NG	**87.4(0.8)** ●	82.1(1.2)	27.9%	1.14x	**88.0(0.5)** ●	0.5%	0.94x	86.6(0.5)	23.2%	1.11x
	OHSUMED	**77.6(1.0)** ▲	70.9(5.9)	45.5%	1.62x	75.3(0.8)	21.9%	1.08x	70.5(0.8)	69.8%	2.14x
	Reuters90	**41.3(2.6)** ●	**41.4(3.6)** ●	50.7%	1.19x	**41.0(1.6)** ●	28.4%	0.69x	**41.0(1.6)** ●	76.9%	0.67x
	WOS-11967	**87.0(0.7)** ●	**85.0(1.2)** ●	45.4%	1.38x	**86.4(0.9)** ●	22.1%	1.06x	84.9(0.6)	68.4%	2.20x
	WebKB	**81.9(2.5)** ●	**76.5(4.9)** ●	42.9%	1.38x	78.5(1.6)	24.1%	1.21x	73.4(2.2)	71.1%	2.56x
	TREC	**94.3(1.1)** ●	92.4(1.1)	31.3%	1.87x	**94.2(1.3)** ●	18.4%	1.65x	**93.6(1.6)** ●	37.8%	2.02x
	WOS-5736	**90.2(0.9)** ▲	87.6(1.0)	50.4%	1.86x	87.4(0.7)	20.1%	1.08x	86.5(1.1)	70.9%	2.89x
Sentiment	SST1	**51.4(1.7)** ●	**42.5(8.7)** ●	18.9%	1.55x	**47.4(9.9)** ●	5.7%	1.44x	**47.4(9.9)** ●	7.7%	0.91x
	pang_movie	**88.2(0.6)** ●	**87.5(0.8)** ●	46.8%	1.83x	**88.3(0.6)** ●	18.8%	1.34x	**87.8(0.4)** ●	63.5%	2.20x
	MR	**86.4(3.3)** ●	56.7(16.1)	46.7%	2.00x	**88.2(1.0)** ●	3.3%	1.29x	38.9(12.3)	48.8%	1.72x
	vader_movie	**90.5(0.4)** ●	**90.4(0.8)** ●	47.2%	1.93x	**90.9(0.8)** ●	18.1%	1.35x	**90.1(0.5)** ●	63.3%	2.10x
	MPQA	**88.6(0.5)** ●	83.7(1.6)	64.2%	1.86x	**88.1(0.4)** ●	11.2%	0.94x	**87.7(0.7)** ●	55.3%	1.52x
	Subj	**96.1(0.5)** ●	**95.8(0.9)** ●	50.8%	2.76x	95.3(0.3)	21.1%	1.67x	**95.4(0.5)** ●	71.2%	3.24x
	SST2	**92.1(0.4)** ●	64.8(9.3)	48.4%	1.99x	**92.6(0.5)** ●	1.9%	1.10x	**92.8(0.6)** ●	5.8%	1.11x
	yelp_reviews	**97.3(0.4)** ●	**97.0(0.3)** ●	58.6%	1.81x	**97.2(0.4)** ●	11.1%	1.12x	**97.3(0.4)** ●	65.3%	2.28x
	vader_nyt	**82.7(1.1)** ●	**82.0(1.8)** ●	39.8%	1.46x	**82.8(1.6)** ●	24.9%	1.29x	**82.4(1.9)** ●	64.6%	2.34x

Table 3.15: Effectiveness, reduction and speedup Analysis. We present for each dataset (row) the MacroF1 results of the application of CNN, LSSm and LSBo IS approaches (columns) considering the **XLnet** classifier. Cells with value in **bold** and with a green background are statistically equivalent to the MacroF1 columns with the higher value (marked as ●or ▲). Furthermore, for each dataset, we present in the cells with orange background color the results with effectiveness up to 5% worse than the higher MacF1 column, respectively.

Based on these results, we now proceed to analyze the results of the application of the three best IS approaches (CNN, LSSm, and LSBo) to these three classifiers (RoBERTa, BART, and XLNet) in terms of the tripod (effectiveness, reduction, and efficiency). Tables 3.13, 3.14 and 3.15 below, present the respective results. As expected we can observe that the IS methods obtained fewer statistical ties than in the previous analysis when we considered the best classifier per dataset. Considering the 5% loss, CNN varies between 12 (BART) and 14 (RoBERTa) equivalent results. LSSm gets equivalent scores in 18 (RoBERTa) and 19 cases (BART and XLNET) while LSBo ranges between 13 (BART) and 14 (RoBERTa and XLNet) ties. Our results also indicate that BART and XLNet are more resilient to reductions with LSSm, obtaining no losses greater than 5% in all datasets. In sum, regardless of the classifier, LSSm obtains good effectiveness scores, though accompanied by high costs in efficiency. Furthermore, as in the analysis considering the best classifier per dataset, we observed that CNN obtains the best tradeoff considering the tripod effectiveness-reduction-efficiency.

An interesting phenomenon that has not been observed in the previous analysis is a small increase in the MacroF1 absolute value of a few cases after the IS reduction, although with no statistical significance. This includes, for instance; (i) RoBERTa with LSSm on 20NG, (ii) BART with both LSSm and LSBo in Reuters90, and (iii) XLNET with LSSm in 20NG and five other sentiment datasets. As these effectiveness improvements come together with gains in terms of reduction and time speedup, especially in Reuters90 (BART) and in the sentiment datasets for XLNet, we hypothesize that the use of IS methods in Transformers-based Text Classifiers may reduce some noise level or at least reduce overfitting. We will investigate this further in the future. Also, other interesting results for the specific classifiers include:

- **RoBERTa**: There are 32 cases (out 57) in which the use of IS methods can maintain statistically equivalent results in terms of MacroF1 when compared with NoSel and with speedup gains in 23 out of these 32 cases; If we consider an acceptable loss of 5%, this changes to 43 speedup gains on 46 effectiveness equivalent to NoSel results.

- **BART**: LSSm and LSBo methods were able to tie with NoSel in terms of MacroF1 in 23 cases. In 12 out of these 23 cases, there were speedup gains. If we consider an acceptable 5% loss, this result improves to 32 ties and 18 speedup gains. The average total speedup of LSSm – the best IS method for BART in terms of effectiveness – was 1.05x (total time), with a maximum of 1.47x.

- **XLNet** was the classifier that most benefited from the application of IS approaches, considering that in 6 of the 19 datasets, the use of one of the selection methods was able to produce higher absolute MacroF1 than NoSel. Note also that in 5 of the 6 results with improved effectiveness, the application of IS methods also achieved a

significant speedup, up to 2.28x. The use of LSSm with XLNet also produced no losses in effectiveness higher than 5% with several speedups.

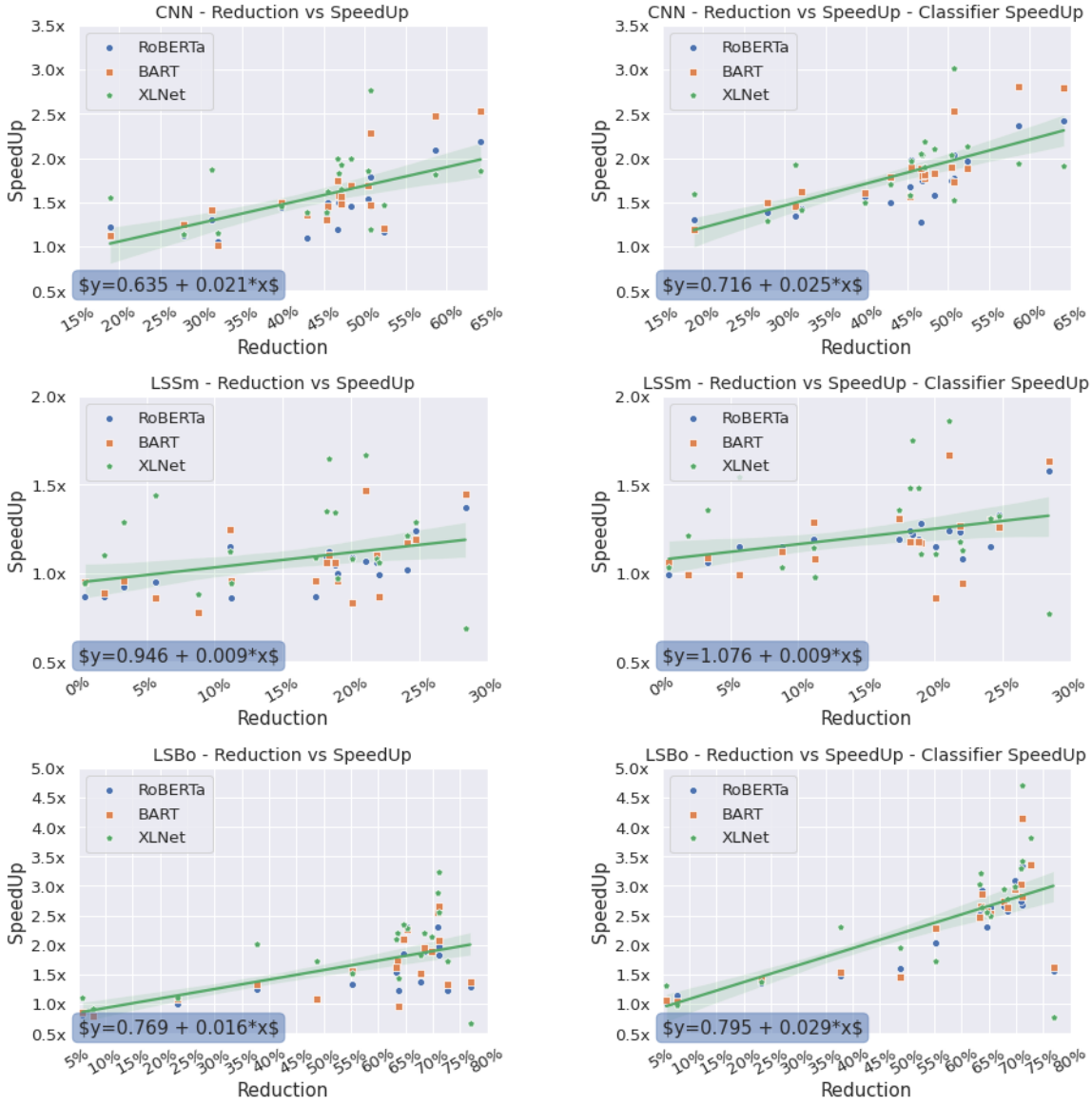
Note that the analyzed results consider the application of the IS methods to the best classifiers. However, one question remains: "Could it be the case that IS improves the results of other methods, perhaps to the point that they can achieve even better results than the ones of the best classifiers?"

A way to answer this question is by performing exhaustive experimentation considering all possible combinations between IS methods and classifiers, and extremely costly procedures. However, considering all results presented in Tables 3.13, 3.14 and 3.15, we can see very similar behaviors, which may indicate a tendency for stability. Thus, is an indication that the results may not change much (or at least not improve) if we experiment with weaker classifiers and weaker IS methods.

3.3.4 Additional Reduction vs. Efficiency Analysis

The graphs in Figure 3.3 present a linear regression analysis of the correlation between the reduction and speedup results obtained by applying CNN, LSSm, and LSBo to the input of ROBERTA, BART, and XLNet. Note that, for each reduction rate in the X-axis obtained on each dataset, we have three associated speedups – one for each classifier. Accordingly, we have 57 points in each graph: 19 datasets times 3 classifiers. The x-axis shows the training set reduction rates and the achieved speedups are shown on the y-axis. The exact values for reduction and speedup can be found in the tables 3.13, 3.14, and 3.15. Finally, we have two versions of the graph for each IS method corresponding to two scenarios: one (a) that considers the time to run the IS method in the total time for building the model as in the previous analysis, and another (b) that does not consider this time. The goal of version (b) is to better evaluate the relationship between speedup and reduction for the Transformers in a scenario in which the cost of running the IS method is nonexistent (zero). In other words, this version captures the direct relation between reduction and speedup for the Transformers without including any extra time imposed by the IS.

Independently of the scenario ((a) or (b)), we notice that all the lines have sublinear behavior and with a large variance across classifiers. The selected training percentages do not reflect linearly in the time of application of the classification model. Concentrating on the scenario that considers the IS time in the model construction (a), we see, for example, that when applying CNN with a reduction of approximately 50%, the speedup is between 1.19x to 2.76x (**XLNet** on Reuters90 and Subj datasets, respectively). For the application of the LSBo with a rate of 70% and 75%, the speedups are between 1.33x



(a) Considering IS method running time

(b) Only construction model time

Figure 3.3: Reduction vs SpeedUp Analysis

(**BART** on DBLP) to 3.24x (**XLNet** on Subj). In sum, our results indicate that the speedups can assume an extensive range of reduction rate values.

Indeed, in some cases in scenario (a), the application of the IS methods may lead to even longer classification times than the model without selection (speedup $< 1.0x$). This behavior is more likely to occur when selection rates are low, for example, in the LSSm method. This occurs due to the overhead caused by using the selection method – in practice, we spend some time generating the selection of instances for the later stage of training the model, but the low reduction rate leads to a time to build the model close to no selection meaning that the additional selection time is not compensated for by model building reduction. Finally, particularly for the Transformers large reduction rates do not guarantee high speedups in the total model construction time. Note, for example,

the application of the LSBo method with a reduction of approximately 65% and 75%. In both cases, the speedup achieved was less than $1.0x$, which indicates that the total time was higher than the model without selection (NoSel).

Finally, when we compared scenarios (a) and (b) we see that there is a slight improvement in the speedup in all cases, especially for LBSo IS method, with a considerable improvement. The higher improvements for LSBo in scenario (b) are due to its good speedup-reduction tradeoff, the second best among all the analyzed IS methods (losing only to CNN), and its higher cost when compared to CNN. If LSSo cost for the time to construct the model is removed, LSBo becomes very competitive when compared to CNN –look at the similarities of the graphs of these two IS methods. This analysis also motivates the construction of cheaper IS methods.

3.3.5 Additional Analysis: Impact of IS on the class distribution

We present in figure 3.4 the impact of applying the three best IS methods in terms of class distribution considering five datasets. On the x-axis, we represent the classes, and on the y-axis the number of instances. The blue bar represents the number of instances without selection (NoSel), and the green bar represents the number of instances after applying the IS method. Also, we have inserted the lines only to help visualize the results.

We concentrate our analysis on the best IS approaches according to the fractional ranking presented for RQ1 (Section 3.3.1): LSSm (Fig. 3.4 - Column 1), CNN (Fig. 3.4 - Column 2), and LSBo (Fig. 3.4 - Column 3). We have chosen these five datasets for three main reasons: (i) they are good representatives of the set of nineteen datasets we use; (ii) they summarize well the behavior of the three IS methods; (iii) they have a low number of classes, which helps a lot the visualization.

To start, note that the **LSSm** IS algorithm maintains the distribution in all cases when comparing distributions with and without the application of the selection method. It is possible to observe in the Figure that the LSSm method prioritizes the minority classes – as seen in the datasets (DBLP, Books, WebKB, and TREC) – where the distributions (with and without selection) are closer. In addition, among the three best IS methods considered in this work, LSSm is the method with the lowest reduction, which is evidenced in the figures. This performance in reduction helps maintain the original distribution. This behavior also helps effectiveness since LSSm was statistically equivalent in 16 of the 19 results presented.

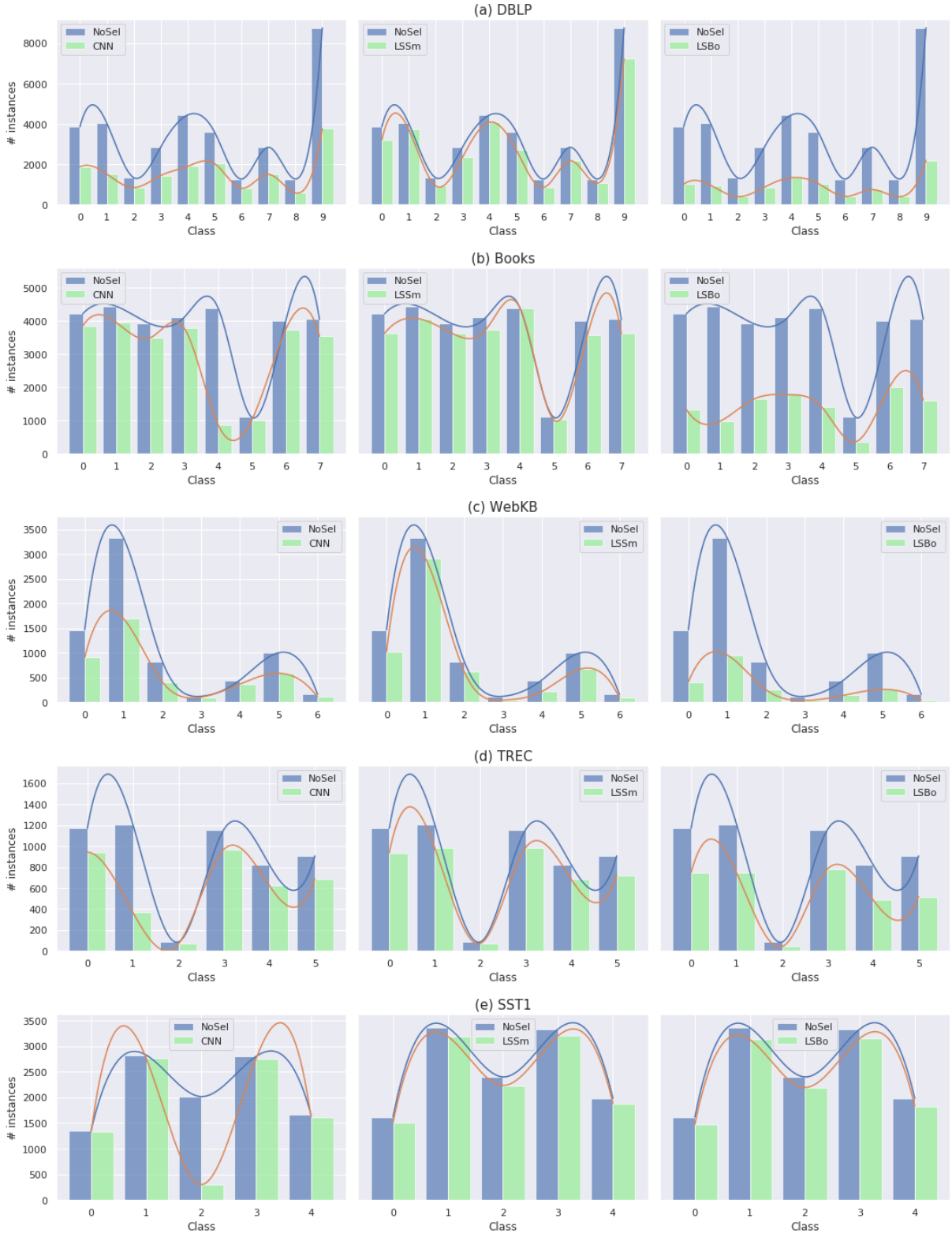


Figure 3.4: The impact of Instance Selection on the class distribution

Now considering **CNN**, it is possible to observe in the SST-1 there was an abrupt reduction in minority classes. The main consequence of changing the distribution, in this case, is significant losses in terms of effectiveness (Table 7). For the other cases, it is possible to observe that CNN maintains the distribution in general, but not in all cases.

Such behavior, i.e., some deviation of the original distribution, can be observed in the CNN application to the Books and DBLP datasets (especially classes 3 to 5 in this last one). Finally, **LSBo** has the highest reduction rate among the top 3 IS methods (56.2%). Notably, in the cases of DBLP and Books, LSBo tends to reduce more than LSSm, while keeping a smoother class distribution than CNN. Furthermore, we can see that it tends to remove (proportionately) more instances of majority classes. This behavior can be seen in classes 1 and 9 of the Books and DBLP datasets.

3.4 Summary

We applied several IS methods to the input data of the best classifiers in each context. Answering RQ1, our evaluation of the tripod constraints (reduction - efficiency - effectiveness), we showed that, in some datasets, specific selection methods can reduce the training set without loss of effectiveness and with efficiency improvements. **However, our experiments revealed that no IS method that can respect all restrictions in all cases.** Also, in some situations, the application of the IS methods can incur additional overheads in the time to construct the model. Our results present a partial answer to the posed question regarding the need for large training sets for performing fine-tuning. In some cases, it is possible to use IS methods to reduce the training set, without loss of effectiveness and efficiency gains – meaning that we do not always need a lot of data. Therefore, these findings neither totally support nor completely refute our posed hypothesis that traditional IS methods are capable of simultaneously respecting all posed restrictions.

CNN	**(P)**: Best overall method, achieving the best trade-off on the tripod Reduction-Effectiveness-Efficiency **(C)**: Limited effectiveness when applied to the largest datasets **(R)**: Medium-to-small topic-related data or sentiment analysis tasks
LSSm	**(P)**: Achieved the best overall Effectiveness **(C)**: Lowest overall Efficiency when considering the three best IS approaches **(R)**: General tasks that cannot deal with effectiveness losses
LSBo	**(P)**: Considering the three best IS methods in terms of effectiveness, achieved the best reduction rate **(C)**: Limited results considering effectiveness when applied to the topic-related tasks **(R)**: Sentiment analysis tasks that need performance and scalability
EGDIS	**(P)**: Considering the IS methods in this list, achieved the best reduction rate and speed up trade-off **(C)**: Limited results considering effectiveness when applied to the medium-to-large topic-related task **(R)**: Tasks that need performance and scalability and may afford some (up to 5%) effectiveness losses
CIS	**(P)**: Good effectiveness results in sentiment analysis tasks **(C)**: Lowest overall Efficiency **(R)**: Not recommended for large NLP tasks due to high associated computational cost
IB3	**(P)**: 4^{th} best IS method considering effectiveness **(C)**: Limited results considering effectiveness when applied to large topic-related tasks **(R)**: Medium-to-small topic-related data or sentiment tasks that can deal with a limited effectiveness loss

Table 3.16: Instance Selection Pros **(P)**, Cons **(C)** and Recommendations **(R)**.

In Table 3.16, we provide pros **(P)**, cons**(C)** and recommendations **(R)** for the six best IS methods according to the Fractional Ranking (Table 3.8) applied to the ATC context. The remaining seven methods were ineffective in the ATC context – according to Table 3.7, they produce statistically worse results in 116 out of 133 (7 methods x 19 datasets). Particularly our focus is on NLP tasks, especially ATC ones.

Our results motivate further investigations on exploiting IS methods in the ATC context, especially regarding new transformers. Our study concerning RQ1 also opens space for designing new, more efficient, effective, and scalable IS methods for the current ATC and the big data scenarios in general. To help close this gap, in the next chapter, we introduce the E2SC framework, which is a new two-step framework that satisfies all the constraints of the tripod and can be used in real-world situations, even with datasets containing thousands of instances, with a special focus on transformer-based architectures.

Chapter 4

An Effective, Efficient, and Scalable Confidence-Based Instance Selection Framework for Transformer-Based Text Classification

In the previous chapter, we found that certain IS methods can reduce the training set size in some datasets without sacrificing effectiveness and even improving efficiency. However, our experiments showed that no one method was able to meet all restrictions in all cases. Additionally, in some scenarios, using these methods increased the time it takes to construct the model, which corresponds to a gap in the literature on methods capable of respecting the posed restriction simultaneously.

To help close this gap, the main contribution of this chapter is the proposal of **E2SC** – **E**ffective, **E**fficient, and **S**calable **C**onfidence-based **I**nstance **S**election – a novel two-step framework[1] aimed at large datasets with a special focus on transformer-based architectures, our first redundancy-oriented IS solution. E2SC is a technique that satisfies the tripod´s constraints and is applicable in real-world scenarios, including datasets with thousands of instances. E2SC´s overall structure can be seen in Figure 4.1.

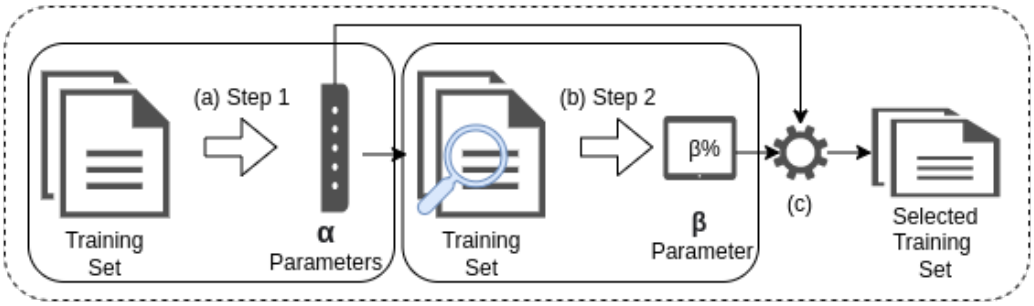


Figure 4.1: The proposed E2SC Framework.

[1]To guarantee the reproducibility of our solution, all the code, the documentation of how to run it and datasets are available on: https://github.com/waashk/e2sc-is/

E2SC´s **first** step – Figure 4.1(a) – aims to assign a probability to each instance being removed from the training set (α parameters). We adopt an exact KNN model solution[2] to estimate the probability of removing instances, as it is considered a **calibrated**[3][110] and computationally cheap (fast) model [21]. Our **first hypothesis (H1)** is that high confidence (if the model is calibrated to the correct class, known in training) positively correlates with redundancy for the sake of building a classification model. Accordingly, we keep the hard-to-classify instances (probably located in the decision border regions), weighted by confidence, for the next step, in which we partially remove only the easy ones.

As the **second** step of our method – Figure 4.1(b) – we propose to estimate a near-optimal reduction rate (β parameter) that does not degrade the deep model's effectiveness by employing a validation set and a weak but fast classifier. Our **second hypothesis (H2)** is that we can estimate the effectiveness behavior of a robust model (deep learning) through the analysis and variation of selection rates in a weaker model. For this, again, we explore KNN. More specifically, we introduce an iterative method that statistically compares, using the validation set, the KNN model's effectiveness without any data reduction against the model with iterative data reduction rates. In this way, we can estimate a reduction rate that does not affect the KNN model's effectiveness. Last, considering the output of these two steps together (Figure 4.1(c)), β% instances are randomly sampled, weighted by the α distribution, to be removed from the training set.

4.1 The Proposed Framework: E2SC

Given a set of instances $X = \{x_1, x_2, ..., x_M\}$, the proposed **E2SC** framework consists of two main steps. The **first** step (Figure 4.1 (a)) aims at estimating a distribution $\alpha(x)$ assigning a probability of x_i being removed from the training set, due to redundancy or lack of informativeness for the sake of constructing a classification model. The **second** step (Figure 4.1 (b)) estimates the β parameter, defined as the near-optimal dataset-specific reduction rate of training instances that does not degrade the model's effectiveness. Considering the output of these two steps together (Figure 4.1(c), β% instances are randomly sampled, weighted by the α distribution, to be removed from the training set. As the main objective of the IS methods is to reduce the computational cost

[2]We depart from the premise that the exact KNN model solution is a reasonable proxy for redundancy. We test and confirm this premise in Sections 4.1.2.1.

[3]A calibrated classifier is one whose probability class predictions correlate well with the classifier´s accuracy, e.g., for those instances predicted with 80% of confidence the classifiers is correct in the prediction is roughly 80% of the cases.

of the most expensive training step, the proposed approach has the following pre-defined constraints: **(i.)** the estimated function f_α must be calibrated and computationally cheap (fast) to learn; and **(ii.)** the beta parameter optimization must be computationally inexpensive to compute and a reasonable estimation of the ideal reduction rate – the one that removes the maximum of instances without degrading the deep model's effectiveness.

As long as both prerequisites are maintained, the E2SC steps´ can be adapted or configured to accommodate different requirements posed by distinct text classification scenarios, given that it can still achieve the reduction, effectiveness and efficiency goals. We present next a first instantiation for both steps of E2SC.

4.1.1 Fitting α Parameters

E2SC first step assigns a probability to each instance being removed from the training set ($\alpha(x)$). The **first hypothesis (H1)** of **E2SC** is that high classification confidence (considering a (weak) calibrated model) positively correlates with redundancy for the sake of building a (strong) classification model. A requirement for this hypothesis is that the chosen weak method for this step must be calibrated (i). In the first E2SC instantiation, we adopt as f the brute-force (exact search) k-nearest neighbor (KNN) model to estimate the probability of removing instances. In Section 4.1.1.1, we partially verify H1 by demonstrating that KNN is a calibrated model. The correlation of confidence with redundancy for model construction will be indirectly captured in the experiments in Section 4.3.1 that aim to answer our RQs. As we shall see, our experiments demonstrate that removing high-confidence predicted instances with KNN does not negatively affect the effectiveness of the Transformer model. Finally, as the main objective of IS is to reduce the total application cost, in Section 4.1.1.1, we demonstrate that KNN is computationally inexpensive for our purposes.

For now, we focus on how we fit the α parameters. The proposed method starts by estimating the α parameters of a probability distribution over a set of distinct classes $\mathcal{Y} = \{y_1, ..., y_c, ..., y_C\}$ given an encoded instance x, as $P(Y = y_c|x) \sim f_\alpha(x)$.

The output of f is probabilities $p1, ..., p_c, ..., p_C$ of each class in $\mathcal{Y}$, where p_c corresponds to the degrees of confidence that f predicted for each class y_c. For the KNN model, the probability p_c of an instance x is given by the ratio between the number of nearest neighbors belonging to class c and the total number of evaluated neighbors (k). The predicted class is $\hat{y} = \text{argmax}_{c \in \{1,..,C\}} f_\alpha(x)$.

The α estimation starts partitioning the instances set into p-folds, containing training and validation splits. The method fits the parameters $f_\alpha(x)^i$ in each fold i using the

training split and applies the adjusted function to predict the text's class in the validation split, generating $P_R(x)^i$. At the end of this step, all instances have been assigned to the y_c class with degrees of confidence p_c. In addition, these training and validation partitions are saved, enabling to perform, in the next stage of E2SC (Section 4.1.2), the iterative statistical comparison correctly, considering the same validation sets.

Thus, considering H1, correctly-predicted instances with higher degrees of confidence can be removed under the assumption that they can be considered redundant for the strong model learning phase. On the other hand, we define the misclassified instances as hard to classify, being kept in the training set, as

$$P_R(x) = \begin{cases} P(y = \hat{y}|x) & \hat{y} == y \\ 0 & \text{otherwise} \end{cases}, \text{ and } y \text{ is } x\text{´s real class.}$$

Next, the $\alpha(x)$ parameters are obtained by **normalizing** $P_R(x)$. Consequently, α can be considered a probability distribution as its sum is up to 1.0. We keep in the training set all the hard-to-classify instances, and, based on the next β parameter optimization (reduction rate), we will partially remove only the easy instances.

4.1.1.1 Hypothesis and Requirement Verification.

The weak model to be adopted by ES2C-IS has to be: (i) calibrated; (ii) efficient, since the main objective of IS is to reduce the total application cost of a robust Transformer-based approach; and (iii) effective, enabling good confidence estimates. Next, we will compare the adopted KNN model to some candidate weak classifiers, including SVM, Random Forest (RF), Naive Bayes (NB), and Nearest Centroid (NC)[4].

H1. Verification Is KNN a calibrated model? If the class prediction probabilities outputted by a classifier have a high correlation with the frequency with which the classifier correctly predicts the instances belonging to that probability range, this classifier is said to be **calibrated** [110]. For example, in instances predicted with 80% confidence, a calibrated classifier is correct in roughly 80% of the cases. As our proposed framework removes instances based on prediction confidence, it is of paramount importance that the adopted classifier be calibrated. We present in Figure 4.2, for the KNN classifier and three datasets used in our experiments (see Section 3.1.1), the distribution between prediction probability ranges (x-axis) and hit-ratio (i.e., correct predictions) (y-axis). It is possible to observe that the KNN fulfills the premise of being a calibrated classifier for all cases. Results with other datasets not shown for space reasons are similar.

[4]For those classifiers, we adopted the same procedures and hyperparameters as in [41].

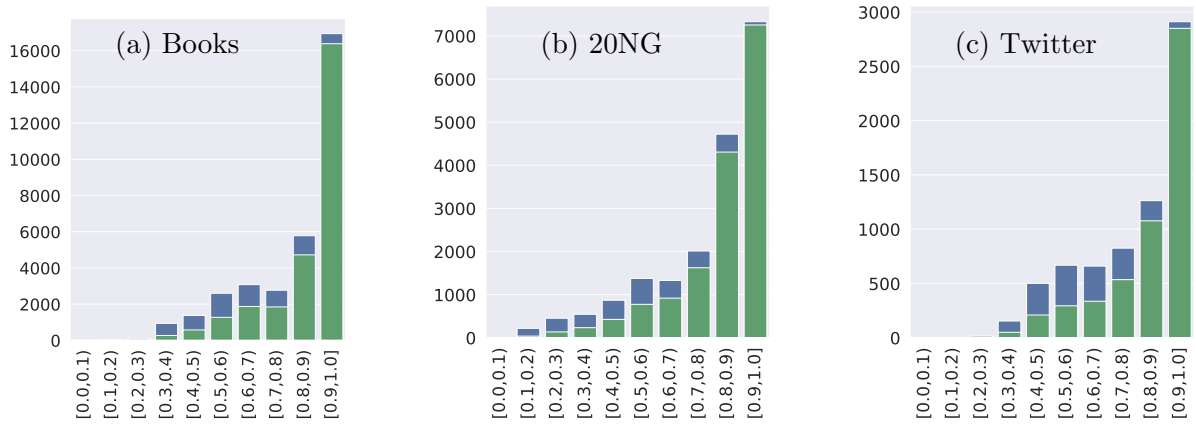


Figure 4.2: Number of instances assigned to each specific range (blue) and the number of correct-predicted instances (green).

To confirm this result, we also analyze the behavior of the weak classifiers using the Brier Score (BS) [11], a scoring rule applied to measure the accuracy of probabilistic predictions, thus, a proper metric to estimate the model calibration. According to [11], this metric is defined as

$$BS = \frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C}(P(Y = y_c|x_i) - o_{ci})^2$$

where o_{ci} is a binary indicator setted to 1 if y_c is x_i' real class, 0 otherwise. BS ranges from 0 to 2 – the closer to zero, the better, achieving more calibrated probability estimations. The obtained BS scores for each candidate weak classifier were: KNN=**0.4**, SVM=0.7, RF=0.5, NB=0.5, and NC=0.8. Based on these, **KNN is the most calibrated** classifier among the considered (weak) ones.

Requirement Verification. Is KNN an efficient model?

As mentioned earlier, the weak model to be adopted by our framework has to be: (i) calibrated; (ii) efficient, since the main objective of IS is to reduce the total application cost of a robust Transformer-based approach; and (iii) effective, enabling good confidence estimates. Achieving these three (potentially conflicting) requirements at the same time is hard, so we hope to choose the classifier with the best tradeoff among them. Table 4.1 presents weak classifier candidates applied to some of the datasets we used in our experiments[5], with their respective results regarding the two remaining aspects: effectiveness and total time.

SVM and RF are the most effective classifiers but have the highest cost (which is consistent with previous works in the literature [31]). When compared to KNN, these strategies are between 19x to 163x slower. Although NB and NC are notably faster than KNN (between 2x and 127x), they have the lowest effectiveness. In the end, KNN is the classifier with the best tradeoff effectiveness-efficiency.

[5]Results with other datasets not shown for space reasons are similar.

	KNN		SVM		RF		NB		NC	
dataset	**Macro-F1**	**Time (s)**	**Macro-F1**	**Time (s)**	**Macro-F1**	**Time (s)**	**Macro-F1**	**Time (s)**	**Macro-F1**	**Time (s)**
Books	81.1(0.5)	157.01	84.1(0.4)	5098.1	74.4(0.5)	3830.8	73.3(0.6)	1.86	62.7(0.7)	1.23
20NG	80.4(0.5)	54.46	89.1(0.7)	2114.8	85.9(0.5)	4615.4	77.4(0.5)	3.44	68.3(0.8)	1.44
ACM	61.3(1.4)	73.86	68.0(0.7)	1434.2	61.7(1.2)	4753.8	40.7(0.8)	3.38	50.5(1.6)	1.11
Twitter	51.2(3.9)	3.04	63.4(1.8)	107.45	38.8(0.6)	497.3	31.4(0.7)	0.40	46.1(1.0)	1.08

Table 4.1: Effectiveness and Efficiency of Weak-Classifiers.

4.1.2 Optimizing the β Parameter

At the end of the first step, all instances have been assigned with an $\alpha(x)$ value. The second step aims at finding the optimal β value, defined as the proportion of instances to remove without degrading the $f_\alpha(x)$ model effectiveness. Our **second hypothesis (H2)** is that we can estimate the effectiveness of a transformer-based model (robust model) through the behavior of the KNN (weak) model by analyzing its selection rate variation.This hypothesis is experimentally verified below (Section 4.1.2.1).

For now, we focus on how we estimate the β parameter. We start by defining β with an initial value $\beta^{(0)}$ and simulate the removal of the corresponding proportion from the training set on each fold weighted by $\alpha(x)$. We then re-estimate $f_\alpha(x^{(\beta)})$ on the shortest training split and measure its effectiveness on the validation split. We then leverage a statistical test (t-test) to compare the effectiveness of $f_\alpha(x)$ and $f_\alpha(x^{(\beta)})$. If they are equivalent, we increment β as follows: $\beta^{(i+1)} = \beta^{(i)} + \delta$. Otherwise, we have already reached the optimal value equal to $\beta^{(i)}$. We repeat this process while the model trained with a fraction of instances remains statistically equivalent to the model trained with the complete instances set. Given that the idea is to iterate as long as it is equivalent, the chosen $f_\alpha(x)$ model must be efficient and reliable to result in an effective cost reduction of the fine-tuning of a robust model.

4.1.2.1 H2 Verification. Can we estimate the effectiveness behavior of a robust model through the behavior of the KNN model?

We verify whether KNN can be used as a weak classifier for this purpose. For this, we generated the correlation between the effectiveness (Macro-F1) of the best classifier

(Transformer) per dataset (Table 4.4) and the effectiveness of KNN. Details of the experimental setup are given in Section 4.2. This result is shown in Figure 4.3. It is possible to visually grasp a very high correlation between KNN and the best Transformer models. The Person's correlation coefficient between the KNN and the best model per dataset is $r = 0.84$.

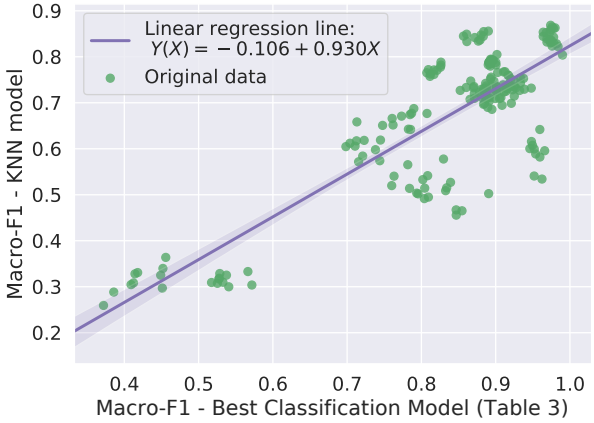


Figure 4.3: Correlation between KNN and Transformers models.

4.1.3 Time Complexity

E2SC complexity is related to the KNN ($\mathcal{O}(N^2)$, where N is the number of instances). In step 1, KNN is applied p times, where p is the number of training-validation partitions. Since p is constant and $p << N$, it is asymptotically dominated by N. In step 2, we run the KNN iteratively to achieve the reduction optimization. Considering both steps, the KNN is applied at most $p \times (\frac{1}{\delta})$ times[6]. In practice, $\frac{p}{\delta}$ is also $<< N$. Therefore, E2SC complexity is $\mathcal{O}(N^2)$).

4.1.4 Model Novelty and Main Contributions

Similarly to IB3 [2], which also belongs to the hybrid category (Section 2.3), **E2SC** chooses the instances that do not negatively affect the model construction if removed, based on whether an auxiliary model classifies them correctly or not. Differently from

[6]As defined in Section 4.4, in our experiments, we fixed the maximum value for $\frac{p}{\delta}$ ratio as 100, but it is usually much smaller than this in practice.

IB3, **E2SC** does not remove the misclassified instances but instead assigns them as hard-to-classify, diminishing their probability of removal from the training for a second stage. In fact, for each correctly-predicted instance, our proposal assigns the probability to be removed proportionally to the KNN confidence prediction. Besides, we propose a near-optimal reduction rate through iterative processes or heuristic-based methods to avoid negatively impacting the deep model's effectiveness. E2SC achieves higher effectiveness at a lower cost (total time) than the current SOTA, as our experiments shall demonstrate.

4.2 Experimental Setup

The experimental setup employed in this chapter closely resembles the one utilized in the previous chapter (Sec 3.1). Thus, we will present it concisely yet comprehensively, emphasizing the incremental and particular modifications implemented in this chapter.

Datasets In **addition** to the datasets present in Table 3.1, we included the topic classification Twitter [4] dataset, in order to study the IS methods' behavior in a naturally-noisier dataset. Also, to demonstrate the flexibility and scalability of E2SC in big data scenarios, we included in our experimentation three new specific datasets with thousands of documents (ranging from 127K to 860K) and different levels of skewness. Table 4.2 shows their statistics.

Dataset	Size	Dim.	# Classes	Density	Skewness
Twitter	6,997	8,135	6	28	Imbalanced
AGNews	127,600	39,837	4	37	Balanced
Yelp_2013	335,018	62,964	6	152	Imbalanced
MEDLINE	860,424	125,981	7	77	Extremely Imbalanced

Table 4.2: New Datasets Statistics

Data Representation and Preprocessing The TFIDF representation is input to all IS methods, including our proposed method. Before creating the TFIDF matrix, we removed stopwords and kept features appearing in at least two documents. We normalized the TF-IDF product result using the L2-norm. In practice, we first construct the TFIDF matrix representation of the documents for the IS stage, and then, we use the corresponding raw document chosen as input for the Transformers classifiers.

Text Classification Methods As mentioned before, our objective in this Chapter is to study and compare our proposed method behavior against the SOTA IS techniques in the context of the recently proposed classification approaches, most notably **Transformer architectures**. Indeed, our ATC comparative investigation conducted in the previous Chapter (Section 3.2) revealed that considering statistical ties the best classifiers were constituted by a Transformer-based method in almost all cases (specifically in 17 out of 19 datasets). In this way, in the next sections, we will consider the best six **transformers-based ATC** methods from the obtained results, namely: **BERT** [44], **RoBERTa** [85], **DistilBERT** [118], **BART** [78], **AlBERT** [73], and **XLNet** [144].

Given a large number of hyperparameters to be tuned, performing a grid search with cross-validation is not feasible for all of them. As a result, to determine the optimum hyperparameter, we applied the same methodology from the previous Chapter. Therefore, we fixed the initial learning rate as $5e-5$, the max number of epochs as 20, and 5 epochs as patience. Finally, we perform a grid search on max_len (150 and 256) and batch_size (16 and 32) since these specified values directly impact efficiency and effectiveness.

	dataset	RoBERTa	BERT	BART	XLNet	DistilBERT	AlBERT
Topic	DBLP	81.4(0.5)	**81.7(0.5)**	81.1(0.5)	81.4(0.6)	81.0(0.6)	77.3(1.0)
	Books	87.2(0.6)	**89.5(0.2)**	86.9(0.5)	87.3(0.4)	87.5(0.5)	84.6(0.8)
	ACM	70.3(1.4)	**71.8(1.0)**	70.8(0.7)	69.9(0.9)	70.1(1.0)	66.2(1.9)
	20NG	86.8(0.7)	85.4(0.5)	87.4(0.9)	**87.4(0.8)**	86.7(0.6)	76.9(1.2)
	OHSUMED	**77.8(1.2)**	76.4(1.2)	77.6(0.7)	77.6(1.0)	76.2(0.7)	66.1(4.8)
	Reuters90	41.9(2.2)	40.2(2.8)	**42.2(2.1)**	41.3(2.6)	40.7(2.5)	41.0(2.6)
	WOS-11967	86.8(0.4)	85.5(0.7)	86.9(0.8)	**87.0(0.7)**	86.0(0.7)	76.8(1.1)
	WebKB	83.0(2.0)	**83.2(2.1)**	83.0(1.7)	81.9(2.5)	82.3(2.1)	80.3(1.4)
	Twitter	78.4(1.8)	64.5(1.9)	**79.0(2.1)**	76.4(2.1)	74.4(2.2)	64.8(2.1)
	TREC	**95.5(0.5)**	87.6(1.4)	95.5(0.8)	94.3(1.1)	95.5(1.1)	93.5(1.4)
	WOS-5736	**90.5(0.9)**	89.7(1.3)	89.6(1.7)	90.2(0.9)	89.2(0.9)	86.7(1.3)
Sentiment	SST1	**53.8(1.3)**	51.6(1.2)	52.8(1.0)	51.4(1.7)	48.9(1.1)	49.2(1.2)
	pang_movie	**89.0(0.4)**	87.4(0.4)	88.1(0.5)	88.2(0.6)	85.2(0.6)	82.9(4.2)
	Movie Review	**89.0(0.7)**	87.7(0.5)	88.2(0.6)	86.4(3.3)	85.2(1.1)	84.9(1.2)
	vader_movie	**91.3(0.5)**	88.2(0.7)	90.4(0.6)	90.5(0.4)	86.6(0.7)	85.4(1.6)
	MPQA	**90.2(0.8)**	89.1(0.7)	90.1(0.7)	88.6(0.5)	88.5(0.6)	87.9(0.6)
	Subj	96.9(0.4)	**97.0(0.3)**	96.8(0.4)	96.1(0.5)	96.0(0.4)	95.5(0.7)
	SST2	**93.2(0.6)**	91.5(0.6)	92.8(0.5)	92.1(0.4)	89.6(0.5)	88.6(2.1)
	yelp_reviews	**97.9(0.4)**	95.6(0.6)	97.5(0.4)	97.3(0.4)	95.6(0.6)	93.9(0.9)
Large	AGNews	**94.2(0.2)**	93.9(0.2)	93.9(0.2)	94.0(0.1)	94.0(0.1)	90.7(0.4)
	Yelp_2013	**64.4(0.6)**	63.6(0.4)	63.8(0.5)	63.0(0.5)	62.3(0.2)	57.2(0.7)
	MEDLINE	81.8(0.6)	75.8(0.8)	**82.2(0.2)**	60.3(0.5)	82.1(0.5)	72.3(3.6)

Table 4.3: Best ATC Approach by Dataset. Results regarding the evaluation metric MacroF1.

We aim to apply the IS methods in the best possible scenario (top-best-ATC-method) for each of the 22 considered datasets. As same as before, we define as the **best approach** (by dataset), the one with the highest effectiveness (MacroF1) among all. We comprehensively and scientifically soundly compared all the aforementioned transformers-

based approaches. This result can be seen in Table 4.3, where the green background corresponds to the best ATC method for each respective dataset. The summary of results of the best approaches by dataset is shown in Table 4.4.

Task	Method	Datasets			
Topic	RoBERTa	OHSUMED	TREC	WOS-5736	AGNews
	BERT	DBLP	Books	ACM	WebKB
	BART	Reuters90	Twitter	MEDLINE	
	XLNet	20NG	WOS-11967		
Sentiment	RoBERTa	SST1	pang_movie	MR	vader_movie
		MPQA	SST2	yelp_reviews	Yelp_2013
	BERT	Subj			

Table 4.4: Summary:Best ATC Approach by Dataset

Instance Selection Methods In this chapter, we consider as baselines the best **six** instance selection methods from the previous chapter (described in Table 3.16), namely: *Condensed Nearest Neighbor* (**CNN**); *Instance Based 3* (**IB3**); *Local Set-based Smoother* (**LSSm**); *Local Set Border Selector* (**LSBo**); *Enhanced Global Density-based Instance Selection* (**EGDIS**); and *Curious Instance Selection* (**CIS**). All parameters for the Instance Selection methods were defined with grid-search, using cross-validation in the training set. Table 3.3 shows the range of parameter values for each IS method we evaluate. The best parameter in each range is marked in **bold**.

Metrics and Experimental Protocol As same as before, all experiments were executed on an Intel Core i7-5820K with 6-Core and 12-Threads, running at 3.30GHz, 64Gb RAM, and a GeForce GTX TITAN X (12GB) and Ubuntu 19.04. We evaluated the classification effectiveness using Macro Averaged F1 (MacroF1)[127] due to skewness in the datasets. We employed the paired t-test with a 95% confidence level to compare the average outcomes from our cross-validation experiments. Finally, we applied the Bonferroni correction [64] to account for multiple tests. We consider reduction mean by defined as $\overline{R} = \frac{\sum_{i=0}^{k} \frac{|T_i|-|S_i|}{|T_i|}}{k}$, where T is the original training set, and S is the solution set containing the selected instances by the IS method being evaluated. Last, in order to analyze the cost-effectiveness tradeoff, we also evaluate each method's cost in terms of the total time required to build the model. The Speedup is calculated as $S = \frac{T_{wo}}{T_w}$, where T_w is the total time spent on model construction using the IS approach, and T_{wo} is the total time spent on execution without the IS phase.

4.3 Experimental Results - Analyses

In this section, we present the results of applying traditional IS methods and our first proposed framework in the context of ATC regarding the RQ2: *Can a novel instance selection method focused on redundancy removal overcome the limitations of existing IS methods to achieve the tripod restrictions in the ATC scenario?*

4.3.1 Is E2SC capable of reducing the training set while keeping classifier effectiveness for each investigated scenario (dataset)?

In these experiments, we consider the premise that the construction time of a deep-learning model is fundamentally related to the amount of training data [36]. In Table 4.5, we present the results regarding the average reduction rate achieved by each selection method. The darker a cell, the larger the reduction achieved by the corresponding method in the respective dataset.

task	dataset	E2SC	CNN	LSSm	LSBo	EGDIS	CIS	IB3
Topic	DBLP	45.0%	52.4%	17.4%	72.8%	62.0%	82.0%	40.0%
	Books	14.0%	32.1%	8.8%	63.7%	62.0%	80.0%	15.0%
	ACM	20.0%	47.1%	19.0%	67.7%	55.0%	46.0%	56.0%
	20NG	21.0%	27.9%	0.5%	23.2%	68.0%	50.0%	5.0%
	OHSUMED	20.0%	45.5%	21.9%	69.8%	57.0%	80.0%	53.0%
	Reuters90	35.0%	50.7%	28.4%	76.9%	54.0%	67.0%	1.0%
	WOS-11967	50.0%	45.4%	22.1%	68.4%	57.0%	77.0%	54.0%
	WebKB	42.0%	42.9%	24.1%	71.1%	53.0%	57.0%	52.0%
	Twitter	35.0%	51.0%	18.0%	70.0%	59.0%	77.0%	60.0%
	TREC	11.0%	31.3%	18.4%	37.8%	39.0%	22.0%	41.0%
	WOS-5736	50.0%	50.4%	20.1%	70.9%	62.0%	69.0%	59.0%
Sentiment	SST1	10.0%	18.9%	5.7%	7.7%	20.0%	60.0%	31.0%
	pang_movie	10.0%	46.8%	18.8%	63.5%	63.0%	77.0%	66.0%
	MR	10.0%	46.7%	3.3%	48.8%	63.0%	58.0%	67.0%
	vader_movie	15.0%	47.2%	18.2%	63.3%	63.0%	75.0%	67.0%
	MPQA	31.0%	64.2%	11.2%	55.3%	45.0%	19.0%	48.0%
	Subj	18.0%	50.8%	21.1%	71.2%	73.0%	51.0%	73.0%
	SST2	15.0%	48.4%	1.9%	5.8%	64.0%	55.0%	68.0%
	yelp_reviews	60.0%	58.6%	11.1%	65.3%	77.0%	60.0%	69.0%
	Average	26.9%	45.2%	15.3%	56.5%	57.7%	61.2%	48.7%

Table 4.5: Percentage of reduction of the training set size.

According to the green scale, CIS, EGDIS, LSBo, and IB3 have the highest reduction rates: on average, 61.2%, 57.7%, 56.5%, and, 48.7%, respectively. The highest reduction rate is for CIS applied to DBLP (82.0%). The lowest reduction rates are obtained by LSSM (on average 15.3%) followed by **E2SC** (26.9%). Thus, considering only the reduction criterion, the first four algorithms stand out. However, the impact on the effectiveness is what, in fact, matters. As we shall see, there is a significant negative impact of the most expressive reductions on effectiveness. In any case, these results show that **all** strategies can reduce the training set size.

The application of the IS methods to the best classifiers in each dataset (Table 4.4) is seen in Table 4.6. The NoSel column corresponds to the results with no training set reduction. We observe in Table 4.6 that **E2SC** is the method that has more statistical ties – 18 datasets (out of 19) – compared to the classification using the complete training set: 10 (out of 11) topic datasets and all sentiment ones. The second best IS approach is LSSm according to this criterion, which was able to maintain the effectiveness levels in 16 cases, followed by CNN – statistically equivalent results in 11 of 19 datasets. Last, CIS, EGDIS, and LSBo (methods with the highest reduction rates) did not perform well, being only able to tie with NoSel in a maximum of 9 different datasets. This demonstrates that excessive reduction is usually detrimental to the Transformer´s effectiveness.

	dataset	NoSel	E2SC	CNN	LSSm	LSBo	EGDIS	CIS	IB3
Topic	**DBLP**	**81.7(0.5)**	79.9(0.6)	79.1(0.8)	**81.1(0.8)**	79.1(0.6)	76.6(0.8)	74.0(1.3)	79.5(0.5)
	Books	**89.5(0.2)**	**89.0(0.3)**	85.9(1.5)	**88.8(0.5)**	84.0(0.5)	84.1(0.6)	80.3(0.5)	72.4(0.4)
	ACM	**71.8(1.0)**	**70.3(1.4)**	67.3(0.8)	**69.6(1.3)**	63.8(1.5)	65.7(1.1)	68.5(1.0)	66.6(0.6)
	20NG	**87.4(0.8)**	**86.3(0.7)**	82.1(1.2)	**88.0(0.5)**	**86.6(0.5)**	79.6(0.4)	81.4(0.9)	82.0(0.4)
	OHSUMED	**77.8(1.2)**	**76.1(1.3)**	73.3(0.4)	73.8(0.5)	68.8(1.2)	67.6(3.3)	61.2(2.0)	71.2(2.0)
	Reuters90	**42.2(2.1)**	**41.8(2.1)**	**42.2(2.0)**	**41.2(2.1)**	**39.8(2.0)**	**42.4(2.6)**	24.1(7.1)	**42.3(2.0)**
	WOS-11967	**87.0(0.7)**	**85.1(0.7)**	**85.0(1.2)**	**86.4(0.9)**	84.9(0.6)	84.3(0.9)	66.1(4.4)	84.7(0.8)
	WebKB	**83.2(2.1)**	**80.9(1.5)**	**81.9(1.6)**	**80.6(1.8)**	76.2(2.1)	**80.5(1.4)**	80.5(1.9)	**80.8(1.8)**
	Twitter	**79.0(2.1)**	**77.6(2.1)**	**77.0(2.3)**	75.3(1.9)	75.9(1.6)	**76.8(2.2)**	73.4(1.6)	**76.9(1.9)**
	TREC	**95.5(0.5)**	**95.3(1.3)**	**94.0(1.0)**	**95.0(0.7)**	**95.0(1.1)**	**92.5(3.2)**	92.4(0.4)	**93.8(1.3)**
	WOS-5736	**90.5(0.9)**	**89.0(1.0)**	**89.2(0.7)**	88.0(1.1)	86.5(1.4)	**88.4(1.3)**	55.4(9.9)	**88.4(1.0)**
Sentiment	**SST1**	**53.8(1.3)**	**52.8(0.7)**	48.0(1.4)	**53.4(0.9)**	**53.2(0.9)**	**53.4(1.0)**	**52.2(0.9)**	**53.3(1.0)**
	pang_movie	**89.0(0.4)**	**88.5(0.6)**	**88.2(0.8)**	**88.5(0.5)**	**88.0(0.6)**	86.8(0.8)	86.9(0.5)	87.1(0.6)
	MR	**89.0(0.7)**	**88.6(0.5)**	**63.6(15.4)**	**89.0(0.6)**	39.3(12.3)	86.5(1.0)	**88.0(0.6)**	87.3(0.8)
	vader_movie	**91.3(0.5)**	**91.1(0.7)**	**90.9(0.5)**	**90.8(0.7)**	**90.5(0.4)**	**89.9(0.6)**	89.1(0.8)	**91.3(0.7)**
	MPQA	**90.2(0.8)**	**89.2(0.9)**	87.0(1.8)	**90.0(0.7)**	**89.9(0.6)**	87.9(0.6)	**90.0(0.7)**	88.7(0.7)
	Subj	**97.0(0.3)**	**96.8(0.3)**	**96.4(0.5)**	95.4(0.7)	95.6(0.5)	96.2(0.4)	**96.7(0.4)**	96.2(0.5)
	SST2	**93.2(0.6)**	**93.1(0.4)**	60.7(11.7)	**92.9(0.5)**	**93.0(0.7)**	91.7(0.7)	92.0(0.8)	92.0(0.8)
	yelp_reviews	**97.9(0.4)**	**97.1(0.4)**	**97.2(0.3)**	**97.7(0.3)**	**97.4(0.3)**	**96.8(0.9)**	**97.3(0.4)**	**97.0(0.5)**

Table 4.6: Macro-F1 - IS approaches (columns) in each dataset (rows) considering the best classifier (Table 3.5). Cells in **bold** and green background are statistically equivalent to no instance selection (**NoSel**).

In sum, both experiments indicate an partial affirmative answer for RQ2 – **E2SC** is capable of reducing the training set while maintaining effectiveness in the vast majority of the cases, achieving the best reduction-effectiveness tradeoff among all methods.

4.3.2 *What is the impact of applying E2SC in the text classification models' total construction time?*

Selecting only the most representative instances should, intuitively, reduce model construction time. By answering the previous question, we demonstrated that **E2SC** reduced the training set while **maintaining** effectiveness. However, adding an IS extra step during the model's pre-construction may cause some time overhead. Indeed, applying an IS method, in some cases, may end up costing even more than building the model with all the data, if the IS step is not cheap enough.

task	dataset	E2SC	CNN	LSSm	LSBo	EGDIS	CIS	IB3
Topic	DBLP	1.26	1.10	0.83	1.11	1.83	0.10	0.68
	Books	1.02	1.04	0.80	1.09	1.91	0.25	0.61
	ACM	1.11	1.44	0.94	1.35	1.94	0.46	1.12
	20NG	1.17	1.35	1.04	1.21	2.49	1.15	0.83
	OHSUMED	1.25	1.49	1.06	1.89	1.58	0.39	1.38
	Reuters90	1.35	1.62	1.22	2.49	1.93	0.96	0.82
	WOS-11967	1.56	1.38	1.06	2.20	2.11	0.87	1.56
	WebKB	1.52	1.39	1.09	2.36	1.63	0.75	1.37
	Twitter	1.27	1.67	0.98	1.89	1.93	0.45	1.66
	TREC	1.07	1.30	1.12	1.24	1.31	0.21	1.23
	WOS-5736	1.58	1.54	1.09	2.30	2.08	1.33	1.78
Sentiment	SST1	1.09	1.22	0.95	0.84	1.21	0.21	0.89
	pang_movie	1.02	1.49	1.05	1.57	2.13	0.53	1.55
	MR	1.03	1.19	0.92	1.09	2.03	0.28	1.53
	vader_movie	1.06	1.59	1.09	1.54	2.12	0.53	0.89
	MPQA	1.19	2.18	0.86	1.33	1.60	0.07	0.85
	Subj	1.14	1.63	1.07	1.72	2.90	0.52	1.87
	SST2	1.06	1.46	0.87	0.81	2.21	0.31	1.80
	yelp_reviews	2.04	2.09	1.15	2.30	3.13	1.45	2.84
	Average	1.25	1.48	1.01	1.60	2.00	0.57	1.33

Table 4.7: SpeedUp on Total Application Cost of the IS Methods applied to the best ATC approach in each dataset.

We consider the total cost as: preprocessing + IS application + training time to build the model. As such, each IS strategy impacts the application time differently. Therefore, for IS methods to be attractive, they must provide efficiency improvements. In Table 4.7, we assess the impact of reducing the training set and if applying IS does compensate in the end for model building. In other words, we compare the Speedups (Sec. 4.2) of each IS approach using the respective (best) classifier for each dataset. We have a color scale for each dataset (row): the greener, the higher speedup; the redder, the higher the computational cost (average execution time) compared to NoSel.

As seen in Table 4.6, **E2SC** achieved excellent effectiveness results and produced attractive training set reductions (on average 26.9%). As we can visually grasp, **E2SC** also achieved satisfactory overall speedup improvements (predominantly light green). The average speed-up for our proposed approach is **1.25** (varying between **1.02** and **2.04**), producing time improvements in **all** scenarios.

CNN has an average speedup of **1.48** – higher than E2SC. However, considering all tripod requirements simultaneously (effectiveness-reduction-efficiency), CNN achieved satisfactory results in just 11 datasets. **LSSm** is the second most costly method (predominantly light green with several red cells). Its low reduction rate, added to its high computational cost, makes the process as a whole not justifiable, given its effectiveness losses. The average speed-up for this approach is **1.01**. The effectiveness losses of **EGDIS** (11 datasets), **LSBo** (10), and **IB3** (11) also make them poor choices, despite the good speedups. Overall, **E2SC** achieved the best tradeoff among all methods, considering all the tripod requirements.

4.3.3 How flexible is the E2SC framework to adjust to different scalability application/task requirements?

Traditional IS strategies do not scale for the big data scenario [36], i.e., datasets with more than 100K instances [150]. In this section, we investigate whether our solution can overcome this barrier and, if not, whether **E2SC** is flexible enough to be adapted to deal with the challenges posed by the task. In other words, we want to demonstrate that proposal's steps can be modified to accommodate different requirements posed by distinct scenarios, mainly those associated with big data.

4.3.3.1 E2SC Framework Instantiation.

Preliminary experiments confirmed that the previously proposed solution did not scale to the new scenarios due to (i) time and (ii) memory consumption restrictions. Time consumption (i) is related to the cost of the iterative near-optimum reduction rate search process. For instance, considering AGNews only, our first instantiation took to select the instances approximately the same time to train the best Transformer with the complete training (no selection). In others words, applying IS would not be viable. The memory

consumption problem (ii) is related to the adoption of the exact KNN solution in the first step of the framework. For instance, according to estimations, considering the largest dataset present in this work (MEDLINE 860K), finding the exact KNN solution would require approximately 2TB of RAM. Thus, to enable the application of our framework in large datasets, we propose two main modifications to our framework.

Modification 1 (M1): Heuristic-Based β Parameter

The first problem is the time spent selecting the instances when a large amount of labeled data is available. Although KNN is relatively computationally cheap, iterating it several times to obtain the optimal beta value can be expensive in large collections – e.g., CIS baseline is based on a weak model (KMeans), but its cost is notoriously high due to a large number of iterations over its weak learner.

Therefore, we propose to modify **E2SC**´s second step, optimizing the parameter β using some heuristics based on the statistical properties of the input dataset. The heuristics comprise two rules. First, we extract two properties of each dataset: document density and a binary feature indicating whether the document class distribution is balanced or not. **These heuristics are based on general observations and lessons learned from the experimental results obtained with the small-to-medium datasets**. First, we observed that: (1) in general, high skewness is detrimental to effectiveness and confidence estimates [43], meaning that we should be more conservative in the reductions for these cases, especially not to harm the smaller classes, whose instances may have lower confidence. We observed that the obtained reduction rate by the automatic iteration in these imbalanced datasets (9 out of 19) was, on average, 28.1%. We consider 25% a conservative approach based on the mean of the results for these datasets (28.1% for imbalanced and 26.9% for all datasets) and the median (31% for this subset and 20% for all datasets).

Second, in balanced datasets, another issue that may affect the effectiveness and induce low confidence in some instances is the lack of data, usually materialized as short documents in the textual datasets. Indeed, we observed that in datasets with less than 100 words per document (low density) – 7 (out of 19), our iterative approach achieved low reductions (between 10% to 21%). In the remaining three balanced and high-density datasets, our approach was able to reduce the data, on average, by half (53%). Based on such empirical evidence, we propose the following rules, which are computed very fast:

Rule 1: if the documents class distribution is imbalanced or extremely imbalanced, then reduce by 25%.

Rule 2: if the documents class distribution is balanced and the average density is low (less than 100), the fixed reduction is 25%. Otherwise, the reduction is 50%.

Modification 2 (M2): Approximated α Parameters

To further scale the application of KNN within our framework, we propose to exploit an approximate KNN solution, more specifically, a strategy that searches for nearest neighbors through the fast approximate nearest neighbor search: **HNSW**[93], a logarithmic complexity solution, implemented in the *nmslib* python package (version 2.1.1). The main question is whether this solution produces (i) good classification results and (ii) good probability estimates.

		Macro-F1		Time (s)	
		Exact	Approximate	Exact	Approximate
Topic	DBLP	77.09(0.69)	76.64(0.66)	44.66	8.12
	Reuters90	31.45(2.10)	30.83(2.15)	8.73	1.29
	WOS-11967	72.68(0.84)	72.38(1.07)	6.01	2.82
Sent.	pang_movie	73.29(1.08)	72.75(1.42)	3.81	0.95
	vader_movie	74.32(0.93)	73.45(1.34)	3.67	0.95
	yelp_reviews	83.65(1.41)	82.76(1.33)	1.20	0.96

Table 4.8: Comparison Exact vs. Approximate KNN

Table 4.8 shows the results of experiments comparing the Macro-F1 using the exact and the approximate KNN (both adopting $k = 10$). In all cases (results are similar in all datasets, not shown due to space constraints), both solutions are statistically equivalent in MacroF1. On the other hand, the approximate solution is between 1.25x to 6.75x faster than the exact one. The second issue, i.e., whether the probabilities estimates are good enough for our goals, will we be assessed indirectly in the experiments described next.

4.3.3.2 Second Instantiation Complexity

Considering M2, the complexity of the first step is reduced to $\mathcal{O}(log(N))$. Furthermore, adopting M1, the second step becomes constant ($\mathcal{O}(1)$). Therefore, considering both modifications, we achieve a logarithmic solution ($\mathcal{O}(log(N))$), feasible for large datasets.

	AGNews							
	NoSel	E2SC#2						
		β=20%	β=25%	β=35%	β=50%	β=65%	β=75%	β=80%	β=85%
Macro-F1	**94.2(0.2)**	**94.0(0.2)**	**93.9(0.2)**	93.7(0.2)	93.2(0.1)	92.6(0.2)	91.3(0.4)	89.6(0.2)	86.6(0.8)
speedUp	-	1.627x	1.708x	2.047x	2.502x	3.610x	4.761x	6.011x	7.476x

	yelp_2013					
	NoSel	E2SC#2				
		β=20%	β=25%	β=30%	β=35%	β=40%
Macro-F1	**64.4(0.6)**	**64.2(0.4)**	**63.8(0.4)**	**63.3(0.1)**	**63.0(0.5)**	**62.4(0.6)**
speedUp	-	1.285x	1.301x	1.445x	1.551x	1.595x

	MEDLINE					
	NoSel	E2SC#2				
		β=20%	β=25%	β=35%	β=50%	β=65%
Macro-F1	**82.2(0.2)**	**81.7(0.3)**	**81.6(0.3)**	**81.2(0.6)**	80.2(0.5)	77.9(0.7)
speedUp	-	1.452x	1.548x	1.781x	2.033x	3.304x

Table 4.9: Reduction-Effectiveness-Speedup Results for E2SC in Large Datasets Scenarios

4.3.3.3 Experimental Results

As in the previous experiments, the E2SC was applied to the best classification approach in each dataset (see Table 4.4)). In Table 4.9, we present the reduction, effectiveness and speedup results. We also present the β reduction rate variation. As before, the NoSel column corresponds to the results with no training set reduction, and **bold** values with green cells correspond to statistically equivalent results to the classifier trained without any selection (NoSel). In Table 4.9, in addition to considering a binary scenario ("statistical tie - (win) vs. loss"), we included a third scenario for analysis, which includes an "acceptable loss", corresponding to a scenario in which a potential reduction in training set size would compensate for the loss in effectiveness. For the sake of simplicity, here we considered a general, arbitrary rate of 5% of loss, which could be different for each dataset and situation [36].

Applying the proposed heuristics rules (Step 2), note that for the 3 datasets, the suggested removal rate is fixed in **25%**. For this reduction rate, the second proposed instantiation – **E2SC#2** – obtained results statistically equivalent to NoSel in **all cases** while producing speedups ranging from **1.301x** (yelp_2013) up to **1.708x** (AGNews).

Note that our method has a fixed beta based on the proposed heuristic (25%), but we evaluate other reduction ratios for the sake of analysis. This analysis demonstrates that the proposed Heuristic-Based β Parameter, despite effective, can be considered somewhat conservative since there is room for further reductions in some datasets without any effectiveness losses, e.g., yelp_2013 and MEDLINE, to up to **40%** and **35%** respectively, with further speedups. In AGNews, our heuristics induced the maximum reduction pos-

sible without any loss. In the future, we will investigate efficient ways to improve our heuristics toward achieving such potential.

Last, also for the sake of analysis, in the scenario of effectiveness losses under 5% compared to NoSel – orange background – **E2SC#2** could increase its reduction rate further (up to **80%** – AGNews), producing even larger speedups - **3.3x** (MEDLINE) and **6.0x** (AGNews).

In sum, the results demonstrate the flexibility of our proposal by modifying its steps to accommodate different requirements in a big data scenario, solidifying its practical applicability.

4.3.3.4 Enhanced Results in Small-to-Medium datasets

We analyze the behavior of **E2SC#2** in the smaller datasets, further demonstrating the flexibility of our solution. In Table 4.10, we present the results regarding our two proposed instantiations of the **E2SC** framework, concerning: (i) the average reduction rate; (ii) Transformer effectiveness (Macro-F1); and (iii) SpeedUps.

As Table 4.10 demonstrates, this second instantiation has an average reduction rate slightly higher than the previous one (28.9%). We also observe that E2SC#2 is statistically equivalent in **all** datasets compared to the classification using the complete training set. As we can visually grasp, **E2SC#2** also achieved satisfactory overall speedup improvements (darker green than the first instantiation). The average **E2SC#2** speedup is higher – **1.37** – producing time improvements in **all** scenarios. This last result demonstrates that the proposed modifications were able to enhance the results in the small-to-medium datasets, considering all constraints.

Indeed, some specific cases are interesting to pinpoint. In both DBLP and Twitter, although the reductions produced by **E2SC#2** were smaller compared to the first instantiation, the speedups were almost the same due to compensations in the overall time produced by the modifications in the IS phase. Moreover, in Reuters90, WOS-11967, and WOS-5736, there were speedup gains despite smaller or equivalent training set reductions, also caused by compensations in time produced by a faster strategy in the IS phase. In these cases, the reductions in time of the IS step obtained with **E2SC#2** were enough to accelerate the speedups, even in the face of smaller reductions.

In sum, both experiments indicate an affirmative answer for RQ2: **E2SC** is *flexible* to adjust to different application requirements, being able to, in all cases, reduce the training set and maintain effectiveness, while providing efficiency improvements.

		Reduction		Effectiveness (Macro-F1)			SpeedUp	
task	dataset	E2SC	E2SC#2	NoSel	E2SC	E2SC#2	E2SC	E2SC#2
Topic	DBLP	45.0%	25.0%	**81.7(0.5)**	79.9(0.6)	**80.7(0.6)**	1.26	1.25
	Books	14.0%	25.0%	**89.5(0.2)**	**89.0(0.3)**	**88.8(0.5)**	1.02	1.29
	ACM	20.0%	25.0%	**71.8(1.0)**	**70.3(1.4)**	**70.2(1.0)**	1.11	1.29
	20NG	21.0%	25.0%	**87.4(0.8)**	**86.3(0.7)**	**86.2(0.8)**	1.17	1.30
	OHSUMED	20.0%	25.0%	**77.8(1.2)**	**76.1(1.3)**	**75.8(1.5)**	1.25	1.34
	Reuters90	35.0%	25.0%	**42.2(2.1)**	**41.8(2.1)**	**43.3(2.6)**	1.35	1.43
	WOS-11967	50.0%	50.0%	**87.0(0.7)**	**85.1(0.7)**	**85.0(0.7)**	1.56	1.96
	WebKB	42.0%	25.0%	**83.2(2.1)**	**80.9(1.5)**	**82.6(2.3)**	1.52	1.33
	Twitter	35.0%	25.0%	**79.0(2.1)**	**77.6(2.1)**	**78.4(2.1)**	1.27	1.28
	TREC	11.0%	25.0%	**95.5(0.5)**	**95.3(1.3)**	**94.9(1.2)**	1.07	1.18
	WOS-5736	50.0%	50.0%	**90.5(0.9)**	**89.0(1.0)**	**89.2(0.8)**	1.58	1.88
Sentiment	SST1	10.0%	25.0%	**53.8(1.3)**	**52.8(0.7)**	**52.4(1.3)**	1.09	1.29
	pang_movie	10.0%	25.0%	**89.0(0.4)**	**88.5(0.6)**	**88.5(0.6)**	1.02	1.26
	MR	10.0%	25.0%	**89.0(0.7)**	**88.6(0.5)**	**88.3(0.7)**	1.03	1.21
	vader_movie	15.0%	25.0%	**91.3(0.5)**	**91.1(0.7)**	**90.8(0.6)**	1.06	1.25
	MPQA	31.0%	25.0%	**90.2(0.8)**	**89.2(0.9)**	**89.4(1.0)**	1.19	1.03
	Subj	18.0%	25.0%	**97.0(0.3)**	**96.8(0.3)**	**96.8(0.3)**	1.14	1.24
	SST2	15.0%	25.0%	**93.2(0.6)**	**93.1(0.4)**	**92.9(0.6)**	1.06	1.20
	yelp_reviews	60.0%	50.0%	**97.9(0.4)**	**97.1(0.4)**	**97.2(0.4)**	2.04	1.98
	Average	26.9%	28.9%	83.53	82.55	82.71	1.25	1.37

Table 4.10: Tripod Results in Small-to-Medium datasets

4.4 Summary

In this chapter, we proposed **E2SC**, a novel **redundancy-oriented** two-step Instance Selection framework aimed at large datasets with a special focus on transformer-based architectures. E2SC brings innovation to the IS field in terms of (i) the exploitation of calibrated weak classifiers (exact and approximate) to estimate the probability of utility of an instance in the training phase of a Transformer and (ii) the introduction of iterative processes and heuristics, learned from an extensive experimental evaluation of IS alternatives, to estimate the ideal reduction rates. Our experiments demonstrated that E2SC can achieve the best results in terms of effectiveness, reduction, and speedup when compared to the current state-of-the-art in the field. Indeed, In our extensive experimental evaluation with 22 datasets, comparing against six SOTA IS baselines and six Transformers classifiers, our final solution managed to reduce the training sets by almost 30% on average while maintaining the same levels of effectiveness in **all** datasets, with speedup improvements of up to 70%. E2SC was also flexible to be adapted to scale to large datasets, which is hard with the baselines. Our results are interesting from both perspectives, theoretical (e.g., Transformers can indeed be trained with fewer data without losing effectiveness) and practical, allowing for savings in energy, budgets, and carbon emissions.

Chapter 5

An Extended Noise-Oriented and Redundancy-Aware Instance Selection Framework for Transformer-Based Automatic Text Classification

In the comparative experiments of Chapter 4, E2SC achieved the best tripod (effectiveness, efficiency, and reduction) results among all the above alternatives. However, as mentioned, E2SC focuses solely on the **removal of redundant instances**, leaving other aspects, such as noise, that may help to further reduce training, untouched. Indeed, as we shall see, in a simulated scenario designed to evaluate the capability of the IS baseline methods and our previous solution to remove noise, none of the IS solutions satisfactorily performed the task. This motivated us to demonstrate the feasibility of proposing a novel extended IS framework capable of removing simultaneously redundant and noisy instances from the training set (RQ3). Next, we will introduce an extended solution to the limitations mentioned earlier. We start by briefly highlighting some open issues from the original solution, proceeding to present our extended bi-objective instance selection solution and how the proposed enhancements (modifications and extensions) address such issues.

5.1 Motivation and Reasoning

Crowdsourcing [125] and soft labeling [116] annotation methodologies are popular solutions for acquiring large amounts of labeled data with reduced costs. These approaches may lead to poor-quality annotations, resulting in noisier data scenarios when compared to manually curated data by domain experts, in which instances in the training set are

assigned to the wrong classes.Indeed, Martins et al. [94] performed a study case evaluating the influence of challenging and noise instances in review domains. The main finding was that users (whether regular individuals or experts) make mistakes in manually classifying complex instances between 56% and 64% of the time. That being true, noise instances can potentially constitute a significant portion of available data in these contexts. Noisy training instances not only have the potential to reduce the effectiveness of the model by introducing misleading patterns, but they may also negatively impact efficiency by requiring additional processing time to extract and incorporate these patterns into the model.

Indeed, preliminary experiments where we artificially inserted noise in the datasets by randomly substituting ground-truth labels with different ones produced significant degradations on the models' effectiveness (up to 4.2% of effectiveness decrease) when 10% of the training become noisy. See Appendix F for more details.

That said, in our previous chapters, we were unable to find cases where effectiveness was improved by the IS methods. Though the datasets we experimented with are well-known benchmarks and very scrutinized by the ATC community, some of them may still contain some level of noise, if not by wrongly assigned labels, perhaps due to outliers. This counter-intuitive phenomenon requires further analysis. As redundancy and noise are orthogonal phenomena, the original redundancy-oriented approach was limited in its noise removal capability. In order to show light in this respect, we carry out an experiment to analyze the noise reduction capability of the IS approaches considered in this work, including E2SC.

5.1.1 Noise removal Capability Experiment

In this experiment, we artificially inserted noise in the datasets by randomly switching the ground-truth label of a fixed percentage of documents (5%)[1] , simulating a prefixed and controlled addition of artificial noise in terms of incorrect labels to the datasets [47]. The main idea of this experiment is to verify the capacity of each IS method in terms of percentual (and absolute) noise removal. The obtained results are presented in Table 5.1.

According to the results, only two methods – LSSm and LSBo – could satisfactorily perform noise reduction, regardless of the task (topic classification or sentiment analysis). More specifically, both approaches achieved an average noise reduction rate between 53% to 61%, i.e., they were capable of removing the percentage of inserted noise from the dataset the respective percentages of instances wrongly labeled by our artificial switching process. The methods were specifically designed with noise removal purposes as they

[1]We have experimented with the following levels of inserted noise: 2.5%, 5%, and 10%. We report 5% only for the sake of conciseness.

dataset	# Inst.	#Noise	E2SC	CNN	LSSm	LSBo	EGDIS	IB3
DBLP	34315	1702	1.47% (25)	3.11% (53)	82.43% (1403)	84.25% (1434)	2.47% (42)	9.46% (161)
Books	30234	1502	0.6% (9)	2.46% (37)	32.29% (485)	34.22% (514)	1.93% (29)	7.66% (115)
ACM	22402	1110	2.97% (32)	4.23% (47)	78.92% (876)	81.35% (903)	3.96% (44)	11.98% (133)
20NG	16954	836	0.36% (3)	3.83% (32)	42.11% (352)	42.11% (352)	0.24% (2)	2.51% (21)
OHSUMED	16471	810	2.1% (17)	2.35% (19)	83.46% (676)	85.06% (689)	2.72% (22)	4.81% (39)
Reuters90	11977	560	0.36% (2)	1.07% (6)	81.25% (455)	81.96% (459)	0.89% (5)	8.75% (49)
WOS-11967	10770	520	0.77% (4)	0.00% (0)	84.04% (437)	84.04% (437)	0.77% (4)	0.96% (5)
WebKB	7376	348	4.89% (17)	9.48% (33)	58.05% (202)	69.83% (243)	14.37% (50)	22.70% (79)
Twitter	6297	246	3.66% (9)	8.13% (20)	78.05% (192)	80.89% (199)	6.50% (16)	26.42% (65)
TREC	5356	258	13.18% (34)	14.34% (37)	45.74% (118)	47.29% (122)	5.81% (15)	24.03% (62)
WOS-5736	5162	252	1.98% (4)	1.59% (4)	85.71% (216)	86.51% (218)	0.79% (2)	4.76% (12)
SST1	10669	522	24.14% (126)	15.71% (82)	7.09% (37)	7.66% (40)	13.41% (70)	26.25% (137)
pang_movie	9594	478	7.53% (35)	23.22% (111)	48.95% (234)	65.48% (313)	26.78% (128)	51.05% (244)
MR	9595	478	5.65% (27)	35.15% (168)	5.65% (27)	52.93% (253)	28.87% (138)	56.69% (271)
vader_movie	9510	470	7.87% (36)	25.11% (118)	46.17% (217)	59.79% (281)	28.94% (136)	54.89% (258)
MPQA	9545	298	6.71% (19)	23.83% (71)	55.03% (164)	58.39% (174)	17.45% (52)	57.72% (172)
Subj	9000	450	4.67% (21)	17.11% (77)	42.44% (191)	54.22% (244)	20.44% (92)	56.44% (254)
SST2	8651	418	7.66% (32)	40.43% (169)	8.85% (37)	10.05% (42)	33.97% (142)	54.31% (227)
yelp_reviews	4500	224	7.59% (17)	13.84% (31)	58.48% (131)	64.73% (145)	23.66% (53)	50.89% (114)
Average			5.48%	12.89%	53.93%	60.6%	12.3%	28.0%

Table 5.1: Artificial Noise Removal Capability Experiment. We present the number of training instances (# Inst.), the number of randomly switched labels (# Noise), the reduction achieved by each approach (Reduction), and the respective noise reduction (Noise Reduction) in percentile and absolute terms.

are based on the concept of local sets and tend to remove "harmful" instances from the training set (See definition in Sec. 2.4). Consequently, instances considered noisy with estimated "harmfulness" score greater than "usefulness" score are removed from the training set.The other methods – CNN, EGDIS, and IB3 – achieved marginal results – between 12.3%-28% of noise removal capability – mainly when applied to the sentiment datasets.

Finally, as expected, the original E2SC framework was not able to satisfactorily remove noise, except in a very specific case where the noise removal rate followed the general removal rate (SST1). **One important factor to mention is that, on average, 89.8% (varying between 74.1% and 99.1%) of the artificially introduced noisy instances were incorrectly predicted by E2SC´s weak classifier.** In other words, these instances had a removal probability assigned equal to zero. This provides additional evidence that noisy instances are highly likely to be considered "hard to classify" and be kept in the training set selected by our specific solution, which also explains the low noise removal capability of E2SC[2]. Moreover, E2SC's weak-classifier (KNN) predicts around 30% of the instances as "hard to classify". This sets up an upper limit on the method's reduction capability to around 70% – a reminder that "hard-to-classify" instances are never removed from the dataset (See more in Appendix E).

In sum, there is an opportunity for further instance removal within the E2SC framework by further exploiting the non-redundant, but potentially noisy, set of hard-to-classify training instances. We exploit this opportunity in the next Section.

[2]Some removal is still expected as a few of the switched instances may be redundant.

5.2 Bi-objective Instance Selection Framework

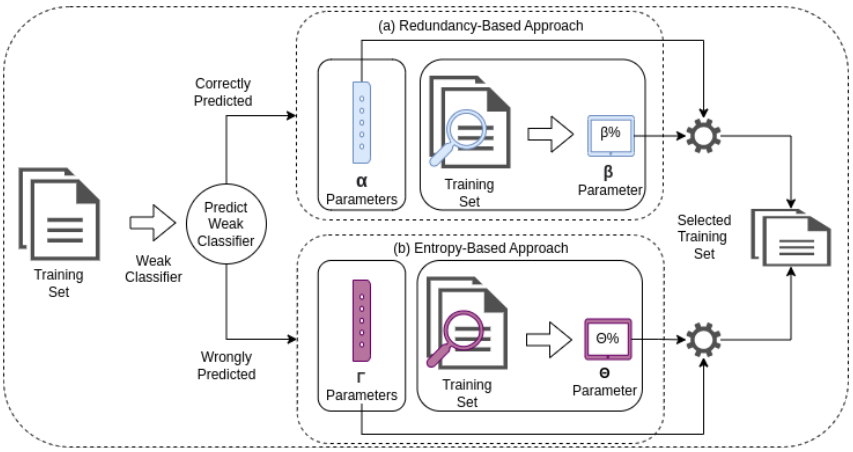


Figure 5.1: Bi-objective Instance Selection Framework

The main contribution of this chapter is the proposal of an *extended bi-objective instance selection (**biO-IS**)* framework built upon our first one aimed at removing both redundant and noisy instances simultaneously. As depicted in Figure 5.1, our extended framework encompasses three main components: a weak classifier, a redundancy-based approach, and an entropy-based approach. In Figure 5.1 (in blue), we depart from our original solution proposed in Chapter 4 considering the Logistic Regression as the calibrated weak classifier instead of KNN, as in our original work. An in-depth comparative analysis of several possibilities of the weak classifiers (Section 5.4.1), including Decision Tree (DT), Logistic Regression (LR), XGBoost, LightGBM (LGBM), and Linear SVM, demonstrated LR as the best option in terms of a trade-off effectiveness-calibration-cost.

To address the second objective of noise removal (the lower part of Figure 5.1 – in purple), we propose a new step based on the entropy and a novel iterative process to estimate near-optimum reduction rates. Considering the instances wrongly predicted by the weak classifier, the main objective is to assign a probability to each of them being removed from the training set based on the probability of the instance being noise. For this, we propose using the entropy function as a proxy to determine the reduction behavior caused by these instances for the sake of training an ATC model. The intuition behind this new step is that when the prediction provided by the calibrated weak classifier is incorrect, the entropy of the posterior probabilities negatively correlates with the confidence of the classifier. This means that low entropy occurs when the classifier assigns an instance with absolute certainty to a wrong class, while high entropy occurs when the classifier is uncertain among several classes. Therefore, we consider the chance of a noisy instance being removed by the inverse of the entropy of the prediction, so when an incorrect prediction is accompanied by low entropy, it is more likely to be removed, and, otherwise, it is more likely to be kept. Accordingly, the proposed biO-IS framework provides a comprehensive solution to address both redundancy and noise removal simultaneously.

5.2.1 Redundancy-based approach

The original[3] E2SC framework[4] consists of two steps. **Step 1** involves estimating a distribution, known as $\alpha(x)$ (alpha), which assigns to each training instance x a probability of being removed from the training set. It is worth noticing this probability distribution is **specifically** focused on removing **redundancy** for the sake of constructing a classification model. **Step 2** involves estimating the beta parameter, defined as the near-optimal dataset-specific reduction rate of training instances that does not degrade the model's effectiveness.

As the main objective of IS is to reduce the cost of applying a strong classification model through smart selection of a reduced data set, both steps of the framework must necessarily be *computationally inexpensive*. Moreover, the alpha probability needs to be *reliable* in terms of modeling the **redundancy** behavior of the instances. Finally, the beta parameter should maximize the trade-off of (high) reduction vs (maintenance) effectiveness.

In the previous chapter, we presented two instantiations of the proposed framework (E2SC#1 and E2SC#2). According to the results presented in Section 4.3.3.4, despite simpler, the best E2SC instantiation in terms of effectiveness, efficiency, and reduction was the second one. Therefore, this second instantiation will be the basis of our new proposal.

5.2.2 Entropy-based approach

In this section, we present our proposal endeavored on noise removal. Analogous to the original framework, our extension consists of two main modules. The first module calculates the probability that a document being removed from the training set, specifically endeavored to removing **noise**, which is represented by the gamma scores (Γ). The second module determines the reduction rate, represented by the theta scores (θ). To improve the selection process, we propose to use the posterior probabilities obtained from the weak classifier for both modules. This approach optimizes the framework's execution and results in a more efficient instance selection phase, compared to our previous proposal, as we shall see.

[3]Represented in the upper portion of Figure 5.1 (in blue).
[4]We call it a framework because it allows different and diverse instantiations, as we shall see.

5.2.2.1 Learning Gamma (Γ) Scores

Considering the instances wrongly predicted by the weak classifier, the goal is to assign a probability to each of them being removed from the training set (Γ) based on the belief of the instance being noise. For this, we propose using the *entropy function* as a proxy to determine the reduction behavior for hard-to-classify instances. Our hypothesis is that hard-to-classify instances with low entropy of the posteriori class distribution can be considered noise for the purpose of building a classification model.

In more detail, given that an instance was wrongly predicted, a low entropy is evidence of high levels of confidence of the weak classifier towards the wrong prediction (thus, likely noise). On the other hand, a high entropy indicates that the weak classifier was not confident when making the prediction, suggesting some "confusion" and that the instance may be useful, especially if it falls near the decision boundary. Those instances may be useful, for instance, to better define decision boundaries. The entropy of the posterior probability distribution of the prediction for hard-to-classify instances provides complementary information with regard to confidence on the prediction for the sake of instance removal.

More formally, the proposed extension starts by analyzing the posterior probability over a set of distinct classes given an encoded instance. We denote this set of classes as $\mathcal{Y} = \{y_1, ..., y_c, ..., y_C\}$, and the a posteriori probability distribution as $P(Y = y_c|x))$ estimated by the weak classifier g. The output of g is a set of probabilities $p_1, ..., p_c, ..., p_C$, where p_c corresponds to the degree of confidence that g predicts for each class y_c. For each instance x, we calculate the value of $\hat{y} = \text{argmax}_{c\in\{1,..,C\}}\, g(x)$. If the predicted class $\hat{y}$ is different from the actual class of x, we assign the value $\Gamma(x) = \log(n) - Entropy(P(Y|x))$ [5] , where $Entropy(Y|x) = -\sum_{c\in C} P(Y = y_c|x) \cdot \log(P(Y = y_c|x))$. After we have assigned the value $\Gamma(x)$ to all instances, we normalize it by the sum of the vector. As a result, the final Γ vector can be treated as a probability distribution since its sum is up to 1.0.

Consider the following binary classification example in Figure 5.2: suppose that x_1 belongs to the positive class (green triangle) and x_2 belongs to the negative class (red ball). The weak classifier incorrectly classified x_1 and x_2 into the respective classes. Specifically, x_1 has posterior probabilities of (0.0, 1.0), resulting in an entropy of 0.0 (low entropy). This indicates that the weak classifier was highly confident when wrongly predicting x_1, and therefore it should have a high probability of being removed ($\Gamma(x) = 1.0 - 0.0 = 1.0$)[6]. Note that x_1 is in the middle of a "cluster" of red balls; this is a strong "spatial" indication that x_1 was either mislabeled or is an outlier - a triangle that "looks like" a circle.

On the other hand, for x_2, the posterior class distribution would be (0.5, 0.5), resulting in the maximum entropy score. This means that the weak classifier was not

[5] The highest entropy score occurs in the uniform distribution: $-\sum_i^n \frac{1}{n}\log(\frac{1}{n}) = -\sum_i^n \frac{1}{n}(\log(1) - \log(n)) = -\frac{1}{n}\sum_i^n -\log(n) = \frac{1}{n}n\log(n) = \log(n)$.

[6] At this point, $\Gamma(x)$ is not normalized yet.

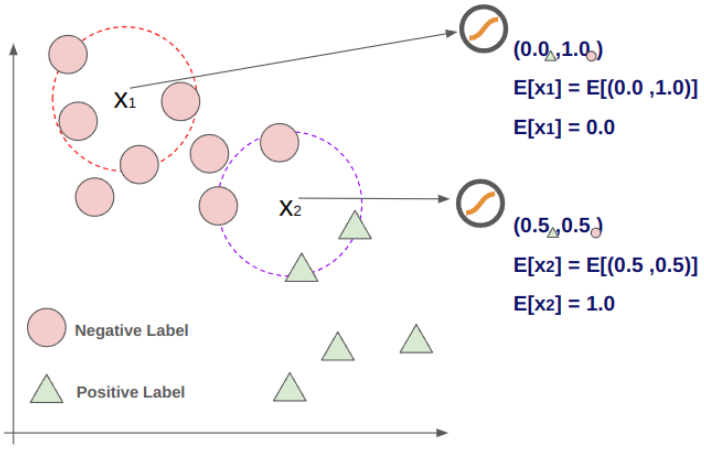


Figure 5.2: Entropy Visual Example

confident when wrongly predicting x_2, and thus it should have a low probability of being removed ($\Gamma(x) = 1.0 - 1.0 = 0.0$)[6]. Notice that x_2 is the frontier of the two classes and keeping it in the training may help to better define the class boundaries[7].

5.2.2.2 Learning Tetha (θ) Score

At the end of the previous step, all instances have been assigned with an $\Gamma(x)$ value, which corresponds to the probability of being kept in the training set. In the context of the original E2SC framework, we have previously demonstrated that there is a strong correlation between the behavior of strong (transformer-based) and weak calibrated classifiers. Thus, similarly to the E2SC´s beta definition step, this step in biO-IS aims at determining the optimal value of θ, representing the proportion of incorrectly predicted instances to be eliminated without compromising the effectiveness of the weak-classifier model.

To this end, we propose a simple yet effective method for defining the value of θ. As the number of incorrectly classified instances, when considering a weak classifier is, on average, smaller than the set of correctly predicted instances (See more in Appendix E), our approach involves initially simulating, based on the Γ distribution learned in the previous step, an aggressive reduction of 50% in the number of "hard-to-classify" instances. We then test two possibilities: (i) removing the instances leads to a degradation in the model's performance, or (ii) the model's effectiveness is maintained after the removal. If the model's performance degrades, we iteratively decrease the reduction rate by a fixed amount ($-\Delta_\Gamma$) until the model performance is not statistically significantly similar to the model trained without data selection. Otherwise, at the beginning, if the model's effectiveness is maintained, we iteratively increase ($+\Delta_\Gamma$) the reduction rate until we observe a model degradation, returning then to the previous iteration's reduction rate. In practice,

[7]For instance, in maximum-margin methods such as SVM x_2 would certainly be a support vector.

starting by removing this percentage rate reduces the number of iterations performed by our proposed approach since, on average, the final θ is around 40%-60% in 90% of the cases.

Finally, combining the two previous steps, we sample θ percent instances from the training set, taking into account the gamma distribution, to be removed.

5.3 Experimental Setup

The experimental setup employed in this chapter closely resembles the one utilized in the previous chapters (Sec 3.1 and Sec 4.2). Thus, we will present it concisely yet comprehensively, emphasizing the particular modifications implemented in this chapter.

Datasets, Data Representation, and Preprocessing To evaluate the IS methods, we adopted the 22 real-world datasets that were previously discussed in the previous chapters (Table 3.1 and Table 4.2), collected from various sources in two broad ATC tasks [80]: i) *topic classification*; and ii) *sentiment analysis*. The TFIDF representation is input to all IS methods, including our proposed method. Before creating the TFIDF matrix, we removed stopwords and kept features appearing in at least two documents. We normalized the TF-IDF product result using the L2-norm. In practice, as illustrated in Figure 3.1, we first split the dataset employing the Stratified K-Fold cross-validation methodology – the smaller datasets were executed using k=10-fold partition, while for the larger ones, we adopted 5 folds due to the cost of the procedure –, then we construct the TFIDF matrix representation of the documents for the IS stage, and then, we use the corresponding raw document chosen as input for the Transformers classifiers.

Text Classification Methods As mentioned, our goal is to study and compare our proposed method against the SOTA IS techniques in the context of **Transformers** architectures – notably the SOTA in classification in several domains[8] [58, 40]. Unlike our previous Chapter, where we selected the best Transformer for each dataset and ran all experiments with a different Transformer. We here employed a more manageable approach. We define a single "averaged" best transformer-based classifier to apply to all datasets based on characteristics such as cross-dataset consistency and reliability in the classification step. Besides reducing the complexity of an already complicated experimental procedure, it also isolates the classifier factor from the analyses of the results. In Section 5.4.2, we compare the effectiveness among the latest version of the following Transformers[9] – **RoBERTa**, **BERT**, **DistilBERT**, **BART**, **AlBERT**, and **XLNet**) – applied to all tested datasets.

[8] LLMs such as GPT and LLama are built on top of Transformer architectures.
[9] We adopted the same hyperparameterization presented in Section 4.2.

Instance Selection Methods In this chapter, we consider as baselines the best **seven** instance selection methods from Chapter 3 (described in Table 3.16), namely: *Condensed Nearest Neighbor* (**CNN**); *Instance Based 3* (**IB3**); *Local Set-based Smoother* (**LSSm**); *Local Set Border Selector* (**LSBo**); *Enhanced Global Density-based Instance Selection* (**EGDIS**); *Curious Instance Selection* (**CIS**); and Effective, Efficient, and Scalable Confidence-Based IS framework (**E2SC**) – our proposal in the previous Chapter. All parameters for the IS methods were defined with grid-search, using cross-validation in the training set. Table 5.2 shows the range of parameter values for each IS method we evaluate. The best parameter in each range is marked in **bold**.

method	parameters
CNN LSSm LSBo	n_neighbors: [**1**, 3, 5, 10]
E2SC	weak-classifier: LR beta: heuristic-based
biO-IS	weak-classifier: LR beta: heuristic-based $\Delta\theta$: 0.1 * θ: pre-fixed for large datasets at 50%

method	parameters
EGDIS	n_neighbors: [1, **3**, 5, 10]
CIS	iterations: 100*|k_cluster| learner: Decision Tree initial error: 0.5 discount factor: 0.01 epsilon: 0.9 to 0.1 (step decay) lr: 0.09 to 0.01 (step decay)
IB3	Confidence Acceptance: 0.9 Confidence Dropping: 0.7

Table 5.2: Parameters of the IS methods.

Metrics and Experimental Protocol In addition to the metrics presented in Section 4.2 – Macro-F1 for measuring effectiveness, Bonferroni correction to assess statistical significance, speedup for measuring efficiency, and reduction – we included a summarization visual concept, in which we summarize our results through an Axial Plot, where we analyze the three imposed restrictions (reduction, effectiveness, and efficiency) simultaneously. More specifically, we normalize the values resulting from each of these metrics (dividing by the respective highest value) and visually summarize the obtained results.

5.4 Experimental Results

In this section, we present the results of applying traditional IS methods and our second proposed framework in the context of ATC regarding the RQ3: *Is it possible to extend the previous proposal to not only remove redundancy but also remove noise, enhancing the level of quality considering all tripod criteria?*

5.4.1 Preliminary Question 1. What is the most suitable weak-classifier to employ within our IS solution?

As both E2SC and biO-IS depend on the definition of a reasonable alternative for the weak classifier (the first step shown in Figure 5.1), we extend the investigation aimed at identifying the best alternative for a weak classifier regarding several posed restrictions. Such analysis has not been performed **extensively** in the previous chapter.

As discussed, our framework selects certain instances to remove based on the predicted posteriori probability of a weak classifier. Our approach seeks to estimate the behavior of a strong classifier by tackling the behavior of a weak classifier as a proxy. To assess the importance of each instance for the final model, the weak classifier should exhibit certain desirable properties, such as calibration, efficiency, and effectiveness. Achieving all three requirements simultaneously is hard, as they may be conflicting. Thus, we aim to select the classifier with the best tradeoff.

As mentioned, we have not thoroughly evaluated and compared various possible weak-classifiers for our IS task. Here we provide a comparison of the following options: Decision Trees (DT), Logistic Regression (LR), XGBoost, LightGBM (LGBM), and Linear SVM. For the sake of completeness, we also consider the set of classifiers previously tested in the previous chapter, which includes KNN (adopted in the E2SC), Random Forest (RF), Naive Bayes (NB), and Nearest Centroid (NC).[10]

Calibration We evaluate the calibration of weak classifiers by means of the Brier Score, (BS) [11] a scoring rule applied to measure the accuracy of probabilistic predictions. Brier [11] defines $BS = \frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C}(P(Y = y_c|x_i) - o_{ci})^2$, where o_{ci} is the one-hot vector with 1 in the index of the true class if x_i, 0 otherwise. BS ranges from 0 (best) to 2 (worst) – the closer to zero, the better in achieving more calibrated probability estimations. Figure 5.3 presents the averaged Brier Score obtained by applying each weak classifier on all datasets considered in our experiments.

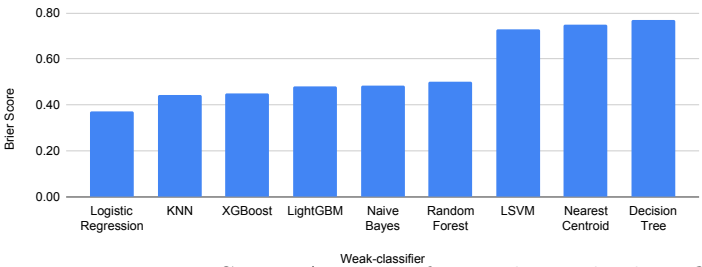


Figure 5.3: Brier Score Average for each weak-classifier

[10]For the sake of reproducibility of these results, the weak-classifier methods' hiperparameterization is available in Appendix G.

Figure 5.3 clearly indicates LR classifier tends to be the better calibrated among the considered weak classifiers.

Efficiency and Effectiveness Trade-off Table 5.3 presents the effectiveness and total time results of the six most calibrated classifiers according to the previous analysis applied to five datasets used in our benchmark [11].

	Logistic Regression		KNN		xgBoost		lightGBM		NB		RF	
dataset	Macro-F1	Time (s)	Macro-F1	Time (s)	Macro-F1	Time (s)	Macro-F1	Time (s)	Macro-F1	Time (s)	Macro-F1	Time (s)
Books	81.2(0.5)	3.82	76.5(0.5)	14.42	75.5(0.5)	41.18	78.8(0.4)	55.46	73.3(0.6)	1.86	75.70(0.6)	154.12
20NG	86.1(0.7)	5.10	82.8(0.4)	5.27	77.8(0.7)	74.83	81.8(0.7)	94.36	77.4(0.5)	3.44	81.64(0.6)	127.07
ACM	59.6(0.6)	1.69	58.6(2.0)	3.43	58.6(0.8)	23.35	62.5(1.5)	23.96	40.7(0.8)	3.38	60.08(0.8)	165.34
Twitter	63.1(1.1)	0.12	52.9(2.2)	0.71	52.9(1.2)	4.05	53.2(2.0)	3.93	31.4(0.7)	0.40	43.59(2.1)	7.97

Table 5.3: Effectiveness and Efficiency of Weak-Classifiers.

On average, Logistic Regression (LR) is the most effective classifier of the list, even when compared to strong competitor approaches like XGBoost, LightGBM, and Random Forest, while, at the same time, resulting in faster times (between 13.8x and 97.8x times faster). LR is comparable in efficiency with respect to Naive Bayes (NB), but much more effective. Compared to K-Nearest Neighbors (KNN), which was the original component of the E2SC framework, LR is both more effective (up to 19%) and efficient (up to 5.9x times faster).

Impact of adopting LR instead of KNN in the original framework (E2SC) Table 5.4 presents the impact of adopting LR instead of KNN in the original framework regarding the trade-off reduction vs effectiveness vs efficiency on the text classification task.

The reduction rate considering both weak classifiers on each dataset is basically the same. This is expected since this rate is based on heuristics regarding two statistical properties of the input dataset - the skewness of the class distribution and document density, as explained in Section 4.3.3.1, therefore, it does not depend on the weak classifier definition. For the specific reduction rates of each dataset, see Table 4.10 - column E2SC#2.

Indeed, the experimental results indicate that the effectiveness of the removal process is better estimated when correlated with the logistic regression confidence instead of the KNN´s confidence. In fact, the reduction performed based on the KNN predictions incurs effectiveness losses in two datasets – Books and OHSUMED. Removal based on LR confidence keeps effectiveness in all datasets

Finally, LR is also more efficient in the selection phase than KNN. Consequently, it provides a better speedup when considering the total time of application (selection + fine-tuning). After all these analyses, it becomes clear the LR is the best choice to be used as weak classifier within our instance Selection framework.

[11] Results obtained with other datasets are similar to those presented in the table.

task	dataset	MacroF1		SpeedUp	
		E2SC (LR)	E2SC (KNN)	E2SC (LR)	E2SC (KNN)
Topic	DBLP	80.5(0.7)	80.5(0.6)	1.314	1.320
	Books	86.5(0.7)	85.5(0.5)	1.362	1.354
	ACM	69.3(1.6)	69.1(1.2)	1.273	1.262
	20NG	85.8(0.8)	85.5(0.6)	1.295	1.344
	OHSUMED	76.5(1.0)	75.8(1.5)	1.354	1.339
	Reuters90	42.6(2.7)	43.1(2.7)	1.296	1.275
	WOS-11967	85.5(0.9)	85.1(0.9)	2.019	2.043
	WebKB	81.0(2.0)	81.0(2.2)	1.160	1.162
	Twitter	77.6(2.6)	77.3(2.7)	1.352	1.298
	TREC	95.1(1.1)	94.9(1.2)	1.230	1.177
	WOS-5736	88.7(1.0)	89.2(0.8)	1.793	1.885
Sentiment	SST1	52.0(0.9)	52.4(1.3)	1.293	1.291
	pang_movie	88.8(0.6)	88.5(0.6)	1.285	1.264
	MR	88.8(0.5)	88.3(0.7)	1.266	1.206
	vader_movie	90.9(0.6)	90.8(0.6)	1.269	1.250
	MPQA	89.7(0.9)	89.4(1.0)	1.176	1.032
	Subj	96.8(0.4)	96.7(0.3)	1.319	1.372
	SST2	92.7(0.5)	92.9(0.6)	1.240	1.197
	yelp_reviews	97.6(0.4)	97.2(0.4)	2.175	1.965
Large	AGNews	93.9(0.2)	93.9(0.2)	1.754	1.708
	Yelp_2013	63.7(0.2)	63.8(0.4)	1.453	1.301
	MEDLINE	81.3(0.5)	81.4(0.4)	1.599	1.548
	Average			1.422	1.391

Table 5.4: Impact of adopting LR instead of KNN in the original framework (E2SC) - Effectiveness vs SpeedUp trade-off – Legend: A red background denotes a effectiveness loss, while a green background indicates a better overall absolute speedup.

5.4.2 Preliminary Question 2. *What transformer-based classifier should we consider for our experimentation?*

In Section 3.2, through a comprehensive and computationally costly set of experiments, we identified, for each dataset, the most effective transformer to be adopted in each dataset. Such strategy is not only impractical and expensive, but also introduces a second factor to be considered besides the IS strategy itself: the interplay between the IS strategy and the transformer.

Differently from the previous chapter, we here rather chose one single "best-on-average" transformer-based classifier to apply to all datasets. Our choice is grounded on two factors: consistency across all datasets and reliability in the classification step.

Table 3.4 shows the effectiveness differences among the latest version of the analyzed transformers (including, RoBERTa [85], BERT [44], DistilBERT [118], BART [78], AlBERT [73], and XLNet [144]) in our benchmark. As we can see, the differences in effectiveness are rather small (between 0.1-2.0 percentual points). In any case, among the tested alternatives, RoBERTa achieves the highest absolute Macro-F1 in 12 out of 22 datasets. In the remaining cases (10 out of 22), RoBERTa performance is not statistically significantly different to the best ATC method per dataset, according to our

t-test at confidence 95% with Bonferroni correction, with marginal differences ranging from 0.10% to 2.09% (0.82% on average), which speaks in favor of its consistency in terms of performance. Therefore, we chose RoBERTa as the "transformer of choice" in the full benchmark, thus factoring out the variability introduced by choosing different architectures and effectively isolating the contribution of the IS technique.

5.4.3 Is biO-IS capable of reducing noisy instances from the training for each investigated scenario?

In this experiment, we again randomly switched the true label of a controled percentage pair of documents, simulating a fixed addition of artificial noise. As performed before, we switched 5% of labels. The main idea of this experiment is to assess the capability of the biO-IS method to remove the type of noise injected by mislabeling.

			E2SC		biO-IS (only entropy-based approach)	
dataset	# Inst.	#Noise	Reduction	Noise Reduction	Reduction	Noise Reduction
DBLP	34315	1702	25.0%	1.47% (25)	12.48%	56.35% (958)
Books	30234	1502	25.0%	0.60% (9)	12.84%	59.72% (896)
ACM	22402	1110	25.0%	2.97% (32)	15.73%	56.67% (629)
20NG	16954	836	25.0%	0.36% (3)	10.09%	63.28% (529)
OHSUMED	16471	810	25.0%	2.10% (17)	15.99%	56.05% (453)
Reuters90	11977	560	25.0%	0.36% (2)	17.73%	66.61% (372)
WOS-11967	10770	520	50.0%	0.77% (4)	14.50%	59.04% (306)
WebKB	7376	348	25.0%	4.89% (17)	18.29%	41.67% (145)
Twitter	6297	246	25.0%	3.66% (9)	13.93%	58.13% (142)
TREC	5356	258	25.0%	13.18% (34)	22.63%	46.9% (120)
WOS-5736	5162	252	50.0%	1.98% (4)	13.13%	53.18% (134)
SST1	10669	522	25.0%	24.14% (126)	34.34%	40.04% (208)
pang_movie	9594	478	25.0%	7.53% (35)	25.00%	47.62% (227)
MR	9595	478	25.0%	5.65% (27)	24.99%	47.62% (227)
vader_movie	9510	470	25.0%	7.87% (36)	25.20%	47.65% (223)
MPQA	9545	298	25.0%	6.71% (19)	32.10%	33.89% (101)
Subj	9000	450	25.0%	4.67% (21)	9.73%	62.67% (282)
SST2	8651	418	25.0%	7.66% (32)	15.33%	47.37% (197)
yelp_reviews	4500	224	50.0%	7.59% (17)	25.00%	47.63% (106)
Average				5.48%		52.21%

Table 5.5: biO-IS - artificial noise removal capability experiment. Legend: In Table, we present the number of training instances (# Inst.), the number of randomly switched labels (# Noise), the reduction achieved by each approach (Reduction), and the respective noise reduction (in percentile and absolute terms).

Table 5.5 presents the results related to the overall removal rate[12] and noise reduction rate for the E2SC and biO-IS approaches (our current proposal). As expected, the

[12]For the E2SC method, we consider the reduction rate provided by its heuristic-based method.

original E2SC framework was not able to satisfactorily remove the artificially introduced noise, except in a very specific case where the noise removal rate followed the general removal rate (SST1). This is likely due to the high correlation between noise and redundancy in this dataset. On average, around 90% (varying between 74% and 99%) of the artificially introduced noisy instances were incorrectly predicted by E2SC's weak classifier, and consequently, these instances are assigned a zero probability to be removed by this method.

When considering only the entropy-based approach (noise-oriented step of our current proposal) that focuses on the removal of non-redundant (but potentially noisy) hard-to-classify training instances, we enhance our original solution capability for noise removal **38.4** times on average (varying between 1.7x and **185.0x**). More specifically, our proposed entropy-based step achieved an average noise reduction rate of 52.2% – varying between 33.9% to 66.6%. When comparing to the results presented in Section 5.1.1, we were capable of satisfactorily removing the manually inserted noise in levels compared to the best baselines (LSSm and LSBo regarding this capability (Table 5.1). In sum, our proposal was able to satisfactorily remove large portions of the artificially inserted noise, demonstrating the potential for the full biO-IS solution.

5.4.4 Is biO-IS capable of reducing the training set while keeping classifier effectiveness for each dataset?

We present in Table 5.6 the impact on the effectiveness of the application of the IS methods to the RoBERTa in each dataset. The NoSel column corresponds to the results with no training set reduction. We stress that the only two IS methods capable of handling large datasets (AGNews, Yelp_2013, and MEDLINE) are ours.

Results reveal that the only IS methods capable of maintaining the effectiveness on all 22 datasets are our methods: biO-IS and E2SC(LR). In other words, the reduction provided by both methods did not harm the classifier's effectiveness.

All IS baseline methods, to a greater or lesser extent, caused losses of effectiveness in one or more datasets. In this sense, regarding this criteria, LSSm can be considered the best baseline, as it was able to maintain effectiveness levels in 14 cases (out of 19 in which it could be run). Following closely behind is CNN, which achieved statistically equivalent results in 11 out of 19 datasets. CIS, EGDIS, and LSBo did not perform as well. They were only able to tie with NoSel in a maximum of 9 different datasets. As we discuss next this is due to their aggressive reduction rates.

We present in Table 5.7 the average reduction rates achieved by each instance selection method (the more intense, the better). According to the green scale, CIS,

task	dataset	NoSel	biO-IS	E2SC (LR)	CNN	LSSm	LSBo	EGDIS	CIS	IB3
Topic	DBLP	81.4(0.5)	**80.5(0.6)**	**80.5(0.7)**	79.0(0.5)	**80.8(0.7)**	78.6(0.9)	74.9(2.4)	73.4(1.4)	78.7(0.5)
	Books	87.2(0.6)	**86.5(0.6)**	**86.5(0.7)**	83.4(1.7)	**86.5(0.6)**	81.5(0.7)	81.3(0.6)	78.9(0.5)	70.6(0.5)
	ACM	70.3(1.4)	**69.0(1.2)**	**69.3(1.6)**	65.4(1.4)	68.0(1.3)	63.4(1.6)	63.5(1.0)	**63.7(6.6)**	64.9(1.3)
	20NG	86.0(0.7)	**85.2(0.7)**	**85.8(0.8)**	81.6(1.1)	**86.9(0.5)**	**85.6(0.6)**	79.3(1.0)	81.3(1.3)	81.9(0.6)
	OHSUMED	77.8(1.2)	**75.1(1.1)**	**76.5(1.0)**	73.3(0.4)	73.8(0.5)	68.8(1.2)	67.6(3.3)	61.2(2.0)	71.2(2.0)
	Reuters90	41.9(2.2)	**40.3(2.1)**	**42.6(2.7)**	**41.7(3.1)**	**41.2(2.1)**	**39.8(2.0)**	**40.5(2.5)**	22.7(7.9)	**42.2(2.2)**
	WOS-11967	86.8(0.4)	**85.6(0.9)**	**85.5(0.9)**	**85.6(0.7)**	**86.5(0.6)**	85.2(0.8)	84.3(0.9)	58.8(5.3)	84.9(0.8)
	WebKB	83.0(2.0)	**80.6(2.1)**	**81.0(2.0)**	**79.8(1.4)**	**78.9(2.1)**	72.5(2.7)	78.3(2.2)	78.5(1.3)	**79.2(1.8)**
	Twitter	78.4(1.8)	**76.5(1.4)**	**77.6(2.6)**	**76.3(1.8)**	75.0(2.0)	71.8(2.3)	**75.4(2.4)**	71.1(2.6)	**75.5(2.0)**
	TREC	95.5(0.5)	**93.3(1.6)**	**95.1(1.1)**	**94.0(1.0)**	**95.0(0.7)**	**95.0(1.1)**	**92.5(3.2)**	92.4(0.4)	**93.8(1.3)**
	WOS-5736	90.5(0.9)	**88.9(1.2)**	**88.7(1.0)**	**89.2(0.7)**	88.0(1.1)	86.5(1.4)	**88.4(1.3)**	55.4(9.9)	**88.4(1.0)**
Sentiment	SST1	53.8(1.3)	**52.8(1.1)**	**52.0(0.9)**	48.0(1.4)	**53.4(0.9)**	**53.2(0.9)**	**53.4(1.0)**	**52.2(0.9)**	**53.3(1.0)**
	pang_movie	89.0(0.4)	**88.2(0.4)**	**88.8(0.6)**	**88.2(0.8)**	**88.5(0.5)**	**88.0(0.6)**	86.8(0.8)	86.9(0.5)	87.1(0.6)
	MR	89.0(0.7)	**88.3(0.4)**	**88.8(0.5)**	**63.6(15.4)**	**89.0(0.6)**	39.3(12.3)	86.5(1.0)	**88.0(0.6)**	87.3(0.8)
	vader_movie	91.3(0.5)	**90.5(0.4)**	**90.9(0.6)**	**90.9(0.5)**	**90.8(0.7)**	**90.5(0.4)**	**89.9(0.6)**	89.1(0.8)	**91.3(0.7)**
	MPQA	90.2(0.8)	**89.0(0.7)**	**89.7(0.9)**	87.0(1.8)	**90.0(0.7)**	**89.9(0.6)**	87.9(0.6)	**90.0(0.7)**	88.7(0.7)
	Subj	96.9(0.4)	**96.0(0.4)**	**96.8(0.4)**	**96.1(0.8)**	95.1(0.5)	95.3(0.4)	**96.2(0.4)**	**96.2(0.3)**	96.1(0.5)
	SST2	93.2(0.6)	**92.4(0.5)**	**92.7(0.5)**	60.7(11.7)	**92.9(0.5)**	**93.0(0.7)**	91.7(0.7)	92.0(0.8)	92.0(0.8)
	yelp_reviews	97.9(0.4)	**97.5(0.3)**	**97.6(0.4)**	**97.2(0.3)**	**97.7(0.3)**	**97.4(0.3)**	**96.8(0.9)**	**97.3(0.4)**	**97.0(0.5)**
Large	AGNews	94.2(0.2)	**94.0(0.2)**	**93.9(0.2)**	-	-	-	-	-	-
	yelp_2013	64.4(0.6)	**64.6(0.2)**	**63.7(0.2)**	-	-	-	-	-	-
	MEDLINE	81.8(0.6)	**81.2(0.4)**	**81.3(0.5)**	-	-	-	-	-	-

Table 5.6: Macro-F1 for different IS approaches (columns) in each dataset (rows) considering RoBERTa as the classifier. Cells in bold and green background highlight results that are not statistically significantly different from those of NoSel.

EGDIS, LSBo, and IB3 are the algorithms with the highest reduction rates – with average reduction rates of 61.2%, 57.7%, 56.5%, and 48.7%, respectively. The topmost reduction rate is for CIS when applied to the DBLP dataset (82.0%). However, as we have seen, these large reduction rates negatively impact effectiveness.

LSSM has the lowest reduction rates, averaging at 15.3%. The original E2SC framework has around 28.4% of reduction rate on average, which, without causing losses of effectiveness in any dataset, can be considered a very good result. We shall remind E2SC is considered the current state-of-the-art in the field. Finally, biO-IS builds on top of ESC-IS, achieving a remarkable 40% reduction rate with no harm at all in effectiveness. biO-IS manages to improve on top of ESC-IS the reduction rate in all datasets, with improvements ranging from 6% up to 128%, in yelp_reviews and yelp_2013, respectively). In all cases but one (Books), biO-IS manages to achieve reduction rates of 30% or higher.

In sum, both experiments indicate an affirmative answer for RQ2 – **biO-IS**, along with **E2SC**, are the only methods capable of significantly (by more than 28%) reducing the training set while maintaining effectiveness in all cases. Moreover, given the improvements of more than 40% on the reduction rate of biO-IS over E2SC, the former is clearly the winner regarding the reduction-effectiveness tradeoff criterion.

task	dataset	biO-IS	E2SC (LR)	CNN	LSSm	LSBo	EGDIS	CIS	IB3
Topic	DBLP	41.0%	25.0%	52.4%	17.4%	72.8%	62.0%	82.0%	40.0%
	Books	29.0%	25.0%	32.1%	8.8%	63.7%	62.0%	80.0%	15.0%
	ACM	37.0%	25.0%	47.1%	19.0%	67.7%	55.0%	46.0%	56.0%
	20NG	31.0%	25.0%	27.9%	0.5%	23.2%	68.0%	50.0%	5.0%
	OHSUMED	34.0%	25.0%	45.5%	21.9%	69.8%	57.0%	80.0%	53.0%
	Reuters90	37.0%	25.0%	50.7%	28.4%	76.9%	54.0%	67.0%	1.0%
	WOS-11967	60.0%	50.0%	45.4%	22.1%	68.4%	57.0%	77.0%	54.0%
	WebKB	35.0%	25.0%	42.9%	24.1%	71.1%	53.0%	57.0%	52.0%
	Twitter	39.0%	25.0%	51.0%	18.0%	70.0%	59.0%	77.0%	60.0%
	TREC	39.0%	25.0%	31.3%	18.4%	37.8%	39.0%	22.0%	41.0%
	WOS-5736	55.0%	50.0%	50.4%	20.1%	70.9%	62.0%	69.0%	59.0%
Sentiment	SST1	55.0%	25.0%	18.9%	5.7%	7.7%	20.0%	60.0%	31.0%
	pang_movie	39.0%	25.0%	46.8%	18.8%	63.5%	63.0%	77.0%	66.0%
	MR	35.0%	25.0%	46.7%	3.3%	48.8%	63.0%	58.0%	67.0%
	vader_movie	37.0%	25.0%	47.2%	18.2%	63.3%	63.0%	75.0%	67.0%
	MPQA	37.0%	25.0%	64.2%	11.2%	55.3%	45.0%	19.0%	48.0%
	Subj	32.0%	25.0%	50.8%	21.1%	71.2%	73.0%	51.0%	73.0%
	SST2	38.0%	25.0%	48.4%	1.9%	5.8%	64.0%	55.0%	68.0%
	yelp_reviews	53.0%	50.0%	58.6%	11.1%	65.3%	77.0%	60.0%	69.0%
Large	AGNews	30.8%	25.0%	-	-	-	-	-	-
	Yelp_2013	57.2%	25.0%	-	-	-	-	-	-
	MEDLINE	31.6%	25.0%	-	-	-	-	-	-
	Average	40.1%	28.4%	45.2%	15.3%	56.5%	57.7%	61.2%	48.7%

Table 5.7: Percentage of reduction of the training set size. Darker cells indicate higher reductions achieved by the corresponding IS method within the dataset.

5.4.5 What is the impact of applying biO-IS in the text classification models' total construction time?

Choosing only the most representative instances should naturally decrease the time required for constructing a model. However, intuitively, including an IS extra step prior to the model's construction phase could potentially result in increased overhead for the overall time. In fact, as we shall discuss, in some cases, incorporating an IS method to reduce the training in the model construction process may require more time than building a model with all the data if the IS step is not sufficiently efficient.

We analyze this behavior in the next experiment, in which we consider the total cost as the sum of the time of preprocessing + IS application + training time to build the model. In other words, for IS methods to be practical, they must also provide efficiency improvements. In Table 5.8, we assess whether reducing the training set with an IS method does pay off for the overall efficacy. To do so, we compare the speedups produced by applying each IS approach to the respective RoBERTa classifier in each dataset. [13]

According to Table 5.8, **biO-IS** (the best selector) can also achieve satisfactory overall speedup improvements. Its average speed-up is **1.67** (varying between **1.31** and **2.46**), producing time improvements in **all** scenarios. E2SC had an average speed-up of

[13]The average total time for model training can be seen in Appendix D

task	dataset	biO-IS	E2SC (LR)	CNN	LSSm	LSBo	EGDIS	CIS	IB3
Topic	DBLP	1.77	1.31	1.17	0.87	1.23	1.94	0.11	0.74
	Books	1.42	1.36	1.05	0.88	1.22	2.04	0.28	0.64
	ACM	1.47	1.27	1.48	1.00	1.37	1.89	0.48	1.18
	20NG	1.42	1.30	1.13	0.87	1.00	2.27	0.99	0.74
	OHSUMED	1.59	1.35	1.49	1.06	1.89	1.58	0.39	1.38
	Reuters90	1.58	1.30	1.46	1.02	2.08	1.81	0.82	0.75
	WOS-11967	2.46	2.02	1.37	0.99	1.89	2.09	0.72	1.74
	WebKB	1.34	1.16	1.10	1.02	1.82	1.49	0.51	1.19
	Twitter	1.43	1.35	1.71	1.02	1.82	2.05	0.40	1.61
	TREC	1.50	1.23	1.30	1.12	1.24	1.31	0.21	1.23
	WOS-5736	2.00	1.79	1.54	1.09	2.30	2.08	1.33	1.78
Sentiment	SST1	2.05	1.29	1.22	0.95	0.84	1.21	0.21	0.89
	pang_movie	1.54	1.28	1.49	1.05	1.57	2.13	0.53	1.55
	MR	1.41	1.27	1.19	0.92	1.09	2.03	0.28	1.53
	vader_movie	1.46	1.27	1.59	1.09	1.54	2.12	0.53	0.89
	MPQA	1.31	1.18	2.18	0.86	1.33	1.60	0.07	0.85
	Subj	1.32	1.32	1.79	1.07	1.98	2.97	0.39	2.30
	SST2	1.42	1.24	1.46	0.87	0.81	2.21	0.31	1.80
	yelp_reviews	2.31	2.17	2.09	1.15	2.30	3.13	1.45	2.84
Large	AGNews	2.02	1.75	-	-	-	-	-	-
	yelp_2013	1.96	1.45	-	-	-	-	-	-
	MEDLINE	1.92	1.60	-	-	-	-	-	-
	Average	1.67	1.42	1.46	0.99	1.54	2.00	0.53	1.35

Table 5.8: SpeedUp on Total Application Cost of the IS Methods applied to RoBERTa in each dataset. The greener, the higher speedup; the redder, the higher the computational cost (average execution time) compared to NoSel.

1.42 (varying between **1.16** and **2.17**). This means significant speedup gains over the state-of-the-art method, achievable due to the extra reduction.

Only one of the strategies presents a slightly higher speedup than biO-IS, – EGDIS with a speedup of 2.0. This method, however, EGDIS achieved satisfactory effectiveness results in only eight datasets. Another strategy that produced a notable speedup is CNN, with an average speedup of 1.46. As EDGIS, CNN achieved satisfactory results in effectiveness in only 11 datasets.

5.4.6 Carbon emissions (CO2e) Considerations

Following, we discuss the emission of CO2 – an estimated measure of greenhouse gases – converted to their equivalent amount of carbon dioxide, which is generated during the IS phase and the classification models' finetuning. This estimation is based on the methodology presented in [74].

This dissertation's performed experiments resulted in about 5600 hours of computation (Section 5.3). Considering [74], we estimate that this computation resulted in approximately 312 kg of CO2e emissions. To put this into perspective, these emissions are

equivalent to driving a distance of 1100 miles in a passenger car or taking four flights from Sao Paulo (Brazil) to Buenos Aires (Argentina). The significant carbon footprint of these experiments highlights the environmental impact of extensive computational tasks often involved in machine learning research. It emphasizes the necessity for more sustainable proposals as biO-IS in developing novel technologies.

Considering the application of RoBERTa in each dataset with no training set reduction (NoSel), the resulting carbon emission amounted to 18.2 kg. On the other hand, when we consider the application of our proposed framework (biO-IS), the resulting carbon emission amounted to only 9.89 kg CO2e. In other words, with emissions of less than 10 kg CO2e (54% of the total emitted), bio-IS would have been able to provide models with equivalent effectiveness to models without the selection stage. This demonstrates the efficacy of biO-IS not only in terms of model performance, training reduction capability, and overall speed-up improvements but also in reducing the environmental impact, serving as the SOTA baseline for the proposal of new IS methods aimed at reducing significantly the cost of ATC models.

5.4.7 Visually summarizing the results

We summarize the results we have obtained considering all the tripod requirements at once. To do so, we normalize the values resulting from each of the considered metrics by dividing them by the respective highest value (reduction and speedup). In the case of effectiveness, the normalized value was the number of datasets in which the IS method reduction caused no effectiveness losses when compared to NoSel. In other words, the reduction did not negatively affect the effectiveness of the classifier (MacroF1).

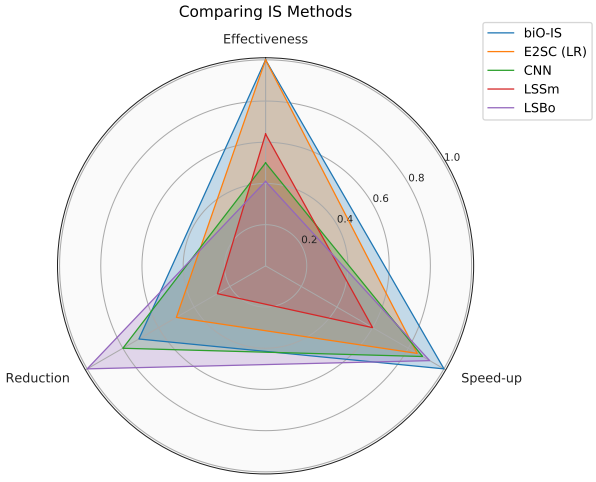


Figure 5.4: Summarizing the results

As shown in Figure 5.4, biO-IS has a balanced performance across all three metrics, offering the best effectiveness and speed-ups while maintaining a good trade-off between reduction and the remaining restrictions. Also E2SC excels in effectiveness but with a reduced reduction rate and speed-up improvements. CNN shows moderate-to-low performance in all metrics. LSSm shows an acceptable effectiveness at the expense of speed-up and reduction. LSBo improves over LSSm regarding reduction and speedup, but it sacrifices the classifier's effectiveness performance in this process.

All in all, **biO-IS** achieved the best tradeoff among all methods, considering all the tripod requirements, considerably advancing the state-of-the-art in IS.

5.5 Summary

In this chapter, we took our research a step further by proposing **biO-IS** – an extended **bi**-**o**bjective **i**nstance **s**election, a novel IS framework aimed at simultaneously removing redundant and **noisy** instances from the training. **biO-IS** estimates redundancy based on scalable, fast, and calibrated weak classifiers and captures noise with the support of a new entropy-based step. We also propose a novel iterative process to estimate near-optimum reduction rates for both steps. Our extended solution is able to reduce the training sets by 41% on average (up to 60%) while maintaining the effectiveness in **all** tested datasets, with speedup gains of 1.67 on average (up to 2.46x). No other baseline, not even our previous SOTA solution, was capable of achieving results with this level of quality, considering the tradeoff among training reduction, effectiveness, and speedup. Moreover, given biO-IS's improvements over E2SC, the former is clearly the winner regarding the reduction-effectiveness tradeoff criterion. Therefore, answering **RQ3**, we demonstrated it was possible to extend the previous proposal to not only remove redundancy but also remove noise, enhancing the level of quality considering all tripod criteria. Indeed, our results are interesting from both theoretical and practical perspectives since we demonstrated that transformers can be effectively trained with less data, leading to energy, budget, and carbon emission savings, **which confirms our Ph.D. dissertation hypothesis (H1).**

Chapter 6

Conclusion and Future Work

In this chapter, we present a summary of the results of this Ph.D. Dissertation and we provide future directions to be pursued after finalizing this dissertation.

6.1 Summary of Results

In this Ph.D. Dissertation, we have surveyed the classical and most recent IS approaches. Our analyses reveal great advances in the last years, with several state-of-the-art methods being proposed and a limited scope of application. Most methods (more than 90%) are applied to small tabular datasets. Applications to NLP are rare, despite being one of the areas that could benefit the most from IS methods.

To help close this gap, we performed a thorough and rigorous comparative study of classical and state-of-the-art IS methods applied to SOTA Automatic Text Classification solutions. In this comparative study, we considered a tripod (reduction-effectiveness-cost) and the respective trade-offs.

Our study was motivated by the massive increase in the cost of new ATC solutions, including the construction of deep learning representations (embeddings) and the huge amount of ever-increasing data available. Specifically, given the success of IS methods in other domains for reducing the amount of training without loss of effectiveness and with gains in efficiency, our work aimed to verify whether such potential benefits extend to the ATC context. This comparative investigation is missing in the literature and constitutes an original contribution of our work.

Based on experiments with over 5,000 measurements, some interesting discoveries have emerged, including:

1. The discovery that, despite the potential of IS methods to deal with noise, this group of methods, in general, considering the best classifier per dataset, was not able to improve the effectiveness of the ATC models, with a few notable exceptions. One such exception is XLNet, in which the use of IS methods generated small improvements

in 6 out of 19 datasets. We hypothesize that this is due to the controlled nature of our datasets (standard benchmarks) but this is yet to be confirmed with further analyses. Results in "wilder" datasets such as those obtained from social network applications where the end user is responsible for the labeling process (instead of a specialist) may produce different results.

2. Among the 13 evaluated IS methods, three IS methods stood out – LSSm, CNN, LSBo –, which have been able to meet all criteria of the tripod (reduction-effectiveness-cost) in 12 out of 19 evaluated datasets – at most. The best IS method (CNN) managed to significantly reduce the training sets (by 46.6% on average) while maintaining the same levels of effectiveness in **12** datasets, with speedups of 1.48 on average.

3. It became clear that in the ATC context, though costly, the fine-tuning step of deep learning networks is crucial for improving effectiveness models. In this context, IS methods can be useful for reducing the costs of building/fine-tuning the models in a non-negligible number of cases (161 out 190 precisely). Indeed in these cases, the IS methods were responsible for reducing the training without incurring in loss of effectiveness, consequently leveraging efficiency improvement. Contrary to what has been widely reported in the literature, we find out that neural networks do not always need a large amount of data to properly work in the ATC context. Representative textual data is enough to achieve high effectiveness at lower costs in some cases.

4. As a secondary contribution, the definition of the current state-of-the-art (neural or non-neural) in 22 textual datasets, extending the work presented in [34];

In sum, our evaluation of the tripod constraints (reduction - efficiency - effectiveness) of several conventional IS techniques indicates that, for the most part, they can reduce the training set size without compromising effectiveness, resulting in improved efficiency. Specifically, when fine-tuning transformer methods, the IS techniques decrease the required amount of data while also delivering substantial reductions in training time. However, there are several cases where traditional approaches fail to meet the tripod requirements fully. It is worth noting that our findings do not entirely support or refute the existing literature on the IS field. Instead, they underscore the need for further research into IS techniques applied to the ATC context, especially with regard to recent transformer architectures. In any case, our investigation highlights a lot of room for developing more efficient, effective, and scalable IS techniques in the big data scenario.

To help close this gap, we also proposed **E2SC**, a novel redundancy-oriented two-step framework that satisfies all the constraints of the tripod and can be used in real-world situations, even with large datasets containing thousands of instances, with a particular focus on transformer-based architectures. E2SC framework introduces two novel approaches to the IS field: (i) the use of calibrated weak classifiers (both exact and approximate) to

estimate the usefulness of data during the training phase of a transformer, and (ii) the incorporation of iterative processes and heuristics, reported by an extensive experimental evaluation of IS alternatives, to determine the optimal reduction rates. The experimental results show that E2SC outperforms the current SOTA in terms of effectiveness, reduction, and speedup. Across 22 datasets, E2SC reduced training sets by almost 30% on average while maintaining effectiveness in all datasets, resulting in speedup improvements of up to 70%. Additionally, E2SC could be adapted to large datasets, which is still challenging for traditional approaches.

Considering the growing adherence to noisy means for dataset annotation (e.g., crowd-sourcing) in relevant scenarios, e.g., social networks where the user herself provides the categorization, it is desired to apply IS approaches considering noise issues. However, as seen in Section 5.1, only two traditional IS approaches could satisfactorily remove noise. The remaining IS approaches demonstrated to be limited in terms of noise removal, including our proposed framework (E2SC). Considering this opens space for developments, we have proposed an extended bi-objective Instance Selection Framework (**biO-IS**), which advances the SOTA in IS by considering removing both redundant and noisy instances simultaneously. **biO-IS** comes as an answer to limitations of our previous work – E2SC – which, as far as we know, was the previous SOTA in IS applied to transformer-based ATC classifiers. Our experimental evaluation considering 22 benchmark datasets applying IS methods as input for the RoBERTa classifier revealed that when evaluating the capability of the IS baseline methods and our previous solution to remove noise, none of the IS solutions performed satisfactorily except for biO-IS, which managed to remove up to 66.6% of the manually inserted noise. **biO-IS** significantly reduced the training sets by 40.1% on average (ranging between 29% and 60% reduction) while maintaining the same levels of effectiveness in all considered datasets. Moreover, **biO-IS** consistently provided speed-ups of 1.67x on average (up to 2.46x). No baseline method achieved results of this quality across all posed criteria. The unique baseline method capable of maintaining effectiveness on all datasets was our previous method E2SC. **biO-IS** outperforms E2SC in reduction rate (41%) and speed-ups (from 1.42x to 1.67x).

These findings have both theoretical and practical implications, **confirming our Ph.D. dissertation hypothesis (H1) by demonstrating that transformers can be trained with fewer data without sacrificing effectiveness, leading to cost and energy savings and, consequently, reduced carbon emissions.**

6.2 Limitations

Large Language Model Considerations Our current work is focused on the application of instance selection methods as a pre-processing step for methods based on 1st and 2nd generation Transformers (e.g., RoBERTa). It should be noted that conducting the experiments presented in this dissertation required a significant amount of time and resources – our experiments sum up around **5,600** hours of continuous computation.

Given the recent rise of state-of-the-art Large Language Models (LLMs) methods, especially *open-sourced* ones such as LLama 3 [131] and Bloom [75], it is quite natural to wonder if and how the proposed instance selection methods would/could be applied to fine-tune these state-of-the-art LLMs for classification and other NLP-relates tasks. The exorbitant cost of fine-tuning these LLM models – between 25-30 times more expensive than fine-tuning 1st and 2nd generation Transformers [117] – makes the application of IS methods very appealing in these scenarios.

However, these enormous costs imply that experiments with such huge models need to be carefully planned to avoid wasting resources. Moreover it is not clear that these very complex LLMs will always be better than the best Transformer in all scenarios. For instance, RoBERTa is a remarkable sentiment classifier [36], often ranking prominently on leaderboards that include sentiment classification, such as the GLUE benchmark[1]. Indeed, results reported in this article in several sentiment datasets are above 90% of macroF1. Even if an LLM can produce further gains of a few percent points, it is doubtful if these gains will translate in *practical improvements* in real-world applications.

Further evidence that the effectiveness gains in specific tasks compensate the much higher costs still need to be provided by the literature. In other words, a conclusive analysis of the cost-benefit is still lacking in the literature, which is essential before we delve into this costly endeavor. We intend to perform the aforementioned cost-benefit analyses soon (and follow the literature for further support). If the results support that the benefits are worth the cost, we will pursue IS with LLMs, perhaps exploiting new computational paradigms such as Quantum Computing [50] to promote scalability.

Further Limitations Despite relevant contributions, our study has some additional limitations, besides the issues regarding the use of LLMs discussed above. Our evaluation targeted specifically the automatic text classification task. Although we have considered a large set of datasets, increasing the number of dataset domains and extending our analysis to include other Information Retrieval tasks such as searching/ranking and recommendation as well as other NLP tasks, for instance, question answering and supervised topic modeling would provide new and valuable insights for the general applicability of IS in a broader scope.

[1] https://gluebenchmark.com/leaderboard/

Despite the general wisdom that the more data, the better the pre-trained model performance, our results motivate a different, less costly approach (i.e., that the better the selected instances, the better the models' performance). Yet, we have not investigated the application of IS in the pre-training stage. In particular, the study of the impact of building a LLM from scratch using an IS framework is an interesting venue to pursue.

6.3 Future Directions

We believe that our work opens up several avenues for investigation. In particular, the exploration of IS methods in many domains has a large potential for improving the conduction and enabling scientific and computationally feasible experiments with large volumes of data, especially for strategies based on deep learning, at a smaller cost and with higher scalability.

In future works, we intend to investigate the introduction of IS techniques in the context of AutoML solutions as a step in a pipeline of transformations as in [37, 31]. In [31], we proposed three new steps – MetaFeatures (MF), Sparsification (SPA), and IS – into the traditional pre-processing phase of pipelines for ATC as well as a thorough and rigorous evaluation of the trade-offs between cost and effectiveness associated with the introduction of these new steps. Therefore, a natural future work would be to investigate the impact of our final solution (biO-IS), as well as, new proposals regarding SPA and MF document representation (e.g., SPLADE [53]) into AutoML pipelines for ATC, aiming to increase their effectiveness while reducing their associated costs.

Considering the vast number of dimensions in textual datasets, assessing how Feature Selection (FS) methods interact with IS can be very interesting. Essentially, this involves working with both the rows (documents) and the columns (features) of a document-feature matrix representing textual data [108]. In this direction, we are already proposing the study of a simple and interesting approach with large potential, which consists of exploring the fact that the document-term matrix (usually coded as TF-IDF) could be seen by vectors of terms instead of the traditional form (vector representation document). In other words, it would be the equivalent of transposing the matrix in question. The main advantages of the proposal are: (i) We do not depend on training and (ii) the possibility of using several FS approaches, already well established in the literature, adapted for IS.

Another challenging issue of the ATC task is how to reduce the costs of obtaining labeled data to train classification models [122]. Solutions to this issue frequently involve active learning (AL) strategies [8], which aim to select, among the often **abundant set of unlabeled data**, only the most "informative" (diversified and representative) data

instances to label. The number of instances selected in stage is typically limited to a given *budget* or the *maximum number of instances* one can afford to label, as the cost and complexity of labeling data are non-negligible. Considering that this scenario is *fully* complementary to instance selection, it would be interesting to study how to merge these approaches in order to take advantage of both scenarios.

Skewness in imbalanced datasets is also an issue for ATC tasks [128]. In this scenario, one (or more) classes are underrepresented, which usually causes a bias in the learning process towards the majority class(es). Usually, the aforementioned issue is handled by applying undersampling solutions [99]. Despite having different objectives, the areas of IS and undersampling are related, as both deal with techniques that aim to select a subset of representative data. Therefore, in future works, we intend to apply the proposed IS methods and compare them with undersampling solutions, aiming to analyze their capability of reducing bias. This scenario consists of ongoing research by our group, in which preliminary results can be seen in [51].

In continual learning (CL) [87], a model is trained incrementally over time on a sequence of datasets, called learning experiences. CL methods should be stable (remember previously learned knowledge), plastic (learn on new data), and efficient (learn quickly and efficiently) even on long timescales and with frequent updates. Most research[30] is focused on improving the stability of DL models. This problem becomes very challenging when past data is unavailable due to catastrophic forgetting (CF): the model quickly forgets past data in some settings. To prevent CF, replay-based methods fine-tune the previous model on the current data and memory. After each learning experience, they update the memory by adding some of the samples from the current experience. When the memory has a fixed size, some of the previous samples are also removed. The application of IS methods to select the most representative training data for CL experiences is a natural one.

Our results motivate investigating the behavior of IS strategies beyond ATC such as searching, ranking, recommendation, and other NLP tasks, such as question answering and supervised topic modeling [138, 60, 23]. More specifically, in the case of searching and ranking, IS can be used to provide initial and potentially more relevant results by delimiting the most representative subset of possible answers, thus improving the overall search experience. In the case of recommendation, more specifically review-aware recommendation systems (RARs), IS approaches may efficiently elucidate the user's preferences by selecting representative reviews to be considered by these approaches. Finally, in the case of topic modeling, our final IS proposal could be used to remove simultaneous redundancy and noise for further effective and efficient TM method application.

Another interesting research line we intend to explore involves applying our proposed framework in the pre-training stage of deep learning models, particularly in scenarios where labeled data is scarce or unavailable. By leveraging our approach, we aim to enhance the efficiency and effectiveness of pre-training (not finetuning) large-scale language

models from scratch. This could involve developing novel techniques for unsupervised or self-supervised learning [84], enabling the model to learn robust and generalizable representations from vast amounts of unannotated text data. The ultimate goal is to improve the performance and scalability of language models while reducing the computational resources and time required for training.

Another interesting research line is enhancing few-shot (FS) [13] learning models. Basically, for this kind of model, a few demonstrations of the task are given at inference time as examples. These examples typically consist of a context and a desired completion. For instance, in the context of sentiment analysis in the product domain, examples could include: "Positive: The camera features are amazing" and "Negative: The product presented durability issues. Not recommend to anyone.". FS approaches involve providing K such examples of context, followed by a final example of context, with the model expected to make the inference. The primary advantage of few-shot learning is a significant reduction in the need for task-specific data, as only a few examples of the context are required. However, a major disadvantage is that the results obtained from this method have generally been inferior to those from SOTA fine-tuned models. Consequently, we are currently proposing the use of IS to enhance the effectiveness of FS approaches.

Last but not least, we intend to perform a cost-benefit analysis on LLMs (effectiveness vs. cost) to potentially pursue IS for fine-tuning LLMs, also exploiting new scalable computational paradigms such as Quantum Computing [50]. In this context, we have preliminary results that exploit Quantum Annealing (QA) [105]. To the best of our knowledge, there have been no prior attempts to tackle the IS problem using QA. We have also proposed a new *Quadratic Unconstrained Binary Optimization* (QUBO) formulation specific for the IS problem. In this line, in future work, we intend to experiment our quantum-based proposal with new QUBO formulations, new transformers, and LLMs. It would also be very interesting to understand the actual environmental impact of quantum annealers. In fact, reducing power and emissions is crucial and there have been attempts to analyze the emissions of several approaches in the IR field [120]. This type of analysis should also be carried out for quantum annealers to understand how much they can impact in providing greener computation.

References

[1] Tariq Abdullah and Ahmed Ahmet. Deep learning in sentiment analysis: A survey of recent architectures. ACM Comput. Surv., jun 2022. Just Accepted.

[2] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. Machine learning, 6(1):37–66, 1991.

[3] Nouf Alturayeif, Hamoud Aljamaan, and Jameleddine Hassine. An automated approach to aspect-based sentiment analysis of apps reviews using machine and deep learning. Automated Software Engineering, 30(2):30, 2023.

[4] Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Leonardo Neves, Vitor Silva, and Francesco Barbieri. Twitter Topic Classification. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[5] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. Modern information retrieval, volume 463. ACM press New York, 1999.

[6] Fatiha Barigou. Impact of instance selection on knn-based text categorization. Journal of Information Processing Systems, 14(2), 2018.

[7] Joran Beel and Bela Gipp. Google scholar's ranking algorithm: the impact of citation counts (an empirical study). In 2009 third international conference on research challenges in information science, pages 439–446. IEEE, 2009.

[8] Fabiano Belém, Washington Cunha, Celso França, Claudio Andrade, Leonardo Rocha, and Marcos André Gonçalves. A novel two-step fine-tuning pipeline for cold-start active learning in text classification tasks. arXiv preprint arXiv:2407.17284, 2024.

[9] Fabiano M Belem, Rodrigo M Silva, Claudio MV de Andrade, Gabriel Person, Felipe Mingote, Raphael Ballet, Helton Alponti, Henrique P de Oliveira, Jussara M Almeida, and Marcos A Goncalves. "fixing the curse of the bad product descriptions"–search-boosted tag recommendation for e-commerce products. Information Processing & Management, 57(5):102289, 2020.

[10] Fabiano Belém, Washington Cunha, Celso França, Claudio Andrade, Leonardo Rocha, and Marcos André Gonçalves. A novel two-step fine-tuning pipeline for cold-

start active learning in text classification tasks. arXiv preprint arXiv:2407.17284, 2024.

[11] Glenn W Brier et al. Verification of forecasts expressed in terms of probability. Monthly weather review, 78(1):1–3, 1950.

[12] Henry Brighton and Chris Mellish. Advances in instance selection for instance-based learning algorithms. Data mining and knowledge discovery, 6(2):153–172, 2002.

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Advances in NeurIPS, volume 33, 2020.

[14] Jose Camacho-Collados and Mohammad Taher Pilehvar. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. arXiv preprint arXiv:1707.01780, 2017.

[15] S. Canuto, D. X. Sousa, M. A. Gonçalves, and T. C. Rosa. A thorough evaluation of distance-based meta-features for automated text classification. IEEE Transactions on Knowledge and Data Engineering (TKDE), 30(12):2242–2256, Dec 2018.

[16] Sergio Canuto, Thiago Salles, Thierson Couto Rosa, and Marcos André Gonçalves. Similarity-based synthetic document representations for meta-feature generation in text classification. In Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pages 355–364, 2019.

[17] Joel Luís Carbonera. An efficient approach for instance selection. In Ladjel Bellatreche and Sharma Chakravarthy, editors, Big Data Analytics and Knowledge Discovery, pages 228–243, Cham, 2017. Springer International Publishing.

[18] Joel Luis Carbonera and Mara Abel. A density-based approach for instance selection. In 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), pages 768–774, 2015.

[19] Joel Luis Carbonera and Mara Abel. A novel density-based approach for instance selection. In 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), pages 549–556, 2016.

[20] Joel Luís Carbonera and Mara Abel. Efficient instance selection based on spatial abstraction. In 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), pages 286–292, 2018.

[21] Thiago NC Cardoso, Rodrigo M Silva, Sérgio Canuto, Mirella M Moro, and Marcos A Gonçalves. Ranked batch-mode active learning. Information Sciences, 379:313–337, 2017.

[22] Pablo Cecilio, Antônio Pereira, Felipe Viegas, Juliana Rosa, Washington Cunha, Fabiana Testa, Elisa Tuler, and Leonardo Rocha. Um framework para extração automática de informações em patentes farmacêuticas. In Anais Estendidos do XXIX Simpósio Brasileiro de Sistemas Multimídia e Web, pages 97–100. SBC, 2023.

[23] Pablo Cecilio, Antônio Perreira, Juliana Santos Rosa Viegas, Washington Cunha, Felipe Viegas, Elisa Tuler, Fabiana Testa Moura de Carvalho Vicentini, and Leonardo Rocha. Patopics: An automatic framework to extract useful information from pharmaceutical patents documents. arXiv preprint arXiv:2408.08905, 2024.

[24] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3):27:1–27:27, 2011.

[25] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. Autodebias: Learning to debias for recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, page 21–30, New York, NY, USA, 2021. Association for Computing Machinery.

[26] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274, 2015.

[27] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. Advances in Neural Information Processing Systems, 33:22243–22255, 2020.

[28] Yu-Ying Chou, Hsuan-Tien Lin, and Tyng-Luh Liu. Adaptive and generative zero-shot learning. In International Conference on Learning Representations, 2021.

[29] Grant Cooper. Examining science education in chatgpt: An exploratory study of generative artificial intelligence. Journal of Science Education and Technology, pages 1–9, 2023.

[30] Andrea Cossu, Antonio Carta, Vincenzo Lomonaco, and Davide Bacciu. Continual learning for recurrent neural networks: an empirical evaluation. Neural Networks, 143:607–627, 2021.

[31] Washington Cunha, Sérgio Canuto, Felipe Viegas, Thiago Salles, Christian Gomes, Vitor Mangaravite, Elaine Resende, Thierson Rosa, Marcos André Gonçalves, and Leonardo Rocha. Extended pre-processing pipeline for text classification: On the role of meta-feature representations, sparsification and selective sampling. Information Processing & Management (IP&M), 57(4):102263, 2020.

[32] Washington Cunha, Celso França, Leonardo Rocha, and Marcos André Gonçalves. Tpdr: A novel two-step transformer-based product and class description match and retrieval method. arXiv preprint arXiv:2310.03491, 2023.

[33] Washington Cunha, Celso França, Guilherme Fonseca, Leonardo Rocha, and Marcos A. Gonçalves. An effective, efficient, and scalable confidence-based instance selection framework for transformer-based text classification. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, New York, NY, USA, 2023. Association for Computing Machinery.

[34] Washington Cunha, Vítor Mangaravite, Christian Gomes, Sérgio Canuto, Elaine Resende, Cecilia Nascimento, Felipe Viegas, Celso França, Wellington Santos Martins, Jussara M. Almeida, Thierson Rosa, Leonardo Rocha, and Marcos André Gonçalves. On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. Information Processing & Management, 58(3):102481, 2021.

[35] Washington Cunha, Leonardo Rocha, and Marcos A Gonçalves. Extended pre-processing pipeline for text classification: On the role of meta-features, sparsification and selective sampling. In Anais Estendidos do XXXVI Simpósio Brasileiro de Bancos de Dados, pages 165–170. SBC, 2021.

[36] Washington Cunha, Felipe Viegas, Celso França, Thierson Rosa, Leonardo Rocha, and Marcos André Gonçalves. A comparative survey of instance selection methods applied to nonneural and transformer-based text classification. ACM Comput. Surv., jan 2023.

[37] Washington Luiz Miranda da Cunha et al. Extended pre-processing pipeline for text classification: on the role of meta-features, sparsification and selective sampling. Universidade Federal de Minas Gerais, 2019.

[38] Monir Dahbi, Saadane Rachid, and Samir Mbarki. Citizen Sentiment Analysis in Social Media Moroccan Dialect as Case Study, pages 16–29. 02 2020.

[39] Spencer Dale et al. Bp statistical review of world energy. BP Plc, London, United Kingdom, pages 14–16, 2021.

[40] Claudio de Andrade, Washington Cunha, Davi Reis, Adriana Silvina Pagano, Leonardo Rocha, and Marcos André Gonçalves. A strategy to combine 1stgen transformers and open llms for automatic text classification. arXiv preprint arXiv:2408.09629, 2024.

[41] Claudio M.V. de Andrade, Fabiano M. Belém, Washington Cunha, Celso França, Felipe Viegas, Leonardo Rocha, and Marcos André Gonçalves. On the class separability of contextual embeddings representations – or "the classifier does not matter when the (text) representation is so good!". Information Processing & Management, 60(4):103336, 2023.

[42] Lingjia Deng and Janyce Wiebe. MPQA 3.0: An entity/event-level sentiment corpus. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1323–1328, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[43] Bart Desmet and Véronique Hoste. Online suicide prevention through optimised text classification. Information Sciences, 439-440:61–78, 2018.

[44] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL), pages 4171–4186, 2019.

[45] Vinicius HS Durelli, Rafael S Durelli, Andre T Endo, Elder Cirilo, Washington Luiz, and Leonardo Rocha. Please please me: does the presence of test cases influence mobile app users' satisfaction? In Proceedings of the XXXII Brazilian Symposium on Software Engineering, pages 132–141, 2018.

[46] Frank Emmert-Streib, Zhen Yang, Han Feng, Shailesh Tripathi, and Matthias Dehmer. An introductory review of deep learning for prediction models with big data. Frontiers in Artificial Intelligence, 3:4, 2020.

[47] Andrea Esuli and Fabrizio Sebastiani. Improving text classification accuracy by training label cleaning. ACM Transactions on Information Systems (TOIS), 31(4):1–28, 2013.

[48] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. Journal of Machine Learning Research (JMLR), 9:1871–1874, June 2008.

[49] Lucas GS Félix, João Victor Silveira, Washington Luiz, Diego Dias, and Leonardo Rocha. Avaliação automática de conteúdo de aplicações de reclamação online. In Anais do VI Symposium on Knowledge Discovery, Mining and Learning, pages 49–56. SBC, 2018.

[50] Maurizio Ferrari Dacrema, Andrea Pasin, Paolo Cremonesi, and Nicola Ferro. Quantum computing for information retrieval and recommender systems. In European Conference on Information Retrieval, pages 358–362. Springer, 2024.

[51] Guilherme Fonseca, Washington Cunha, and Leonardo Rocha. Análise comparativa de métodos de undersampling em classificação automática de texto baseada em transformers. Revista Eletrônica de Iniciação Científica em Computação, 22(1), 2024.

[52] Guilherme Fonseca, Gabriel Prenassi, Washington Cunha, Marcos André Gonçalves, and Leonardo Rocha. Estratégias de undersampling para redução de viés em classificação de texto baseada em transformers. In Proceedings of the 30th Brazilian Symposium on Multimedia and the Web, 2024.

[53] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. From distillation to hard negative sampling: Making sparse neural ir models more effective. In Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval, pages 2353–2359, 2022.

[54] Celso França, Rennan C. Lima, Claudio Andrade, Washington Cunha, Pedro O. S. Vaz de Melo, Berthier Ribeiro-Neto, Leonardo Rocha, Rodrygo L. T. Santos, Adriana Silvina Pagano, and Marcos André Gonçalves. On representation learning-based methods for effective, efficient, and scalable code retrieval. Neurocomputing, 2024.

[55] Salvador Garcia, Joaquin Derrac, Jose Cano, and Francisco Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE transactions on pattern analysis and machine intelligence, 34(3), 2012.

[56] Salvador Garcia, Joaquin Derrac, Jose Cano, and Francisco Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(3):417–435, 2012.

[57] Siddhant Garg, Goutham Ramakrishnan, and Varun Thumbe. Towards robustness to label noise in text classification via noise modeling. CoRR, abs/2101.11214, 2021.

[58] Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. A survey on text classification algorithms: From text to predictions. Inf., 13:83, 2022.

[59] Andrew B. Goldberg, Xiaojin Zhu, and Stephen Wright. Dissimilarity in graph-based semi-supervised classification. In Marina Meila and Xiaotong Shen, editors, Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, volume 2 of Proceedings of Machine Learning Research, pages 155–162, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR.

[60] Christian Reis Fagundes Gomes, Felipe Augusto Resende Viegas, Washington Luiz Miranda da Cunha, and Leonardo Chaves Dutra da Rocha. Cluwords: Explorando clusters semânticos entre palavras para aprimorar modelagem de tópicos. Revista Eletrônica de Iniciação Científica em Computação, 17(2), 2019.

[61] Xiao Han, Yuqi Liu, and Jimmy Lin. The simplest thing that can possibly work:(pseudo-) relevance feedback via text classification. In Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, 2021.

[62] Peter Hart. The condensed nearest neighbor rule (corresp.). IEEE transactions on information theory, 14(3):515–516, 1968.

[63] William Hersh, Chris Buckley, T. J. Leone, and David Hickam. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In Bruce W. Croft and C. J. van Rijsbergen, editors, SIGIR '94, pages 192–201, London, 1994. Springer London.

[64] Yosef Hochberg. A sharper bonferroni procedure for multiple tests of significance. Biometrika, 75(4), 1988.

[65] David Hull. Using statistical testing in the evaluation of retrieval experiments. In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pages 329–338, 1993.

[66] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. In Programming with TensorFlow, pages 87–104. Springer, 2021.

[67] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. An introduction to statistical learning, volume 112. Springer, 2013.

[68] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In Proceedings of the Conference European Chapter Association Computational Linguistics (EACL), pages 427–431, 2017.

[69] Antônio Pereira De Souza Júnior, Pablo Cecilio, Felipe Viegas, Washington Cunha, Elisa Tuler De Albergaria, and Leonardo Chaves Dutra Da Rocha. Evaluating topic modeling pre-processing pipelines for portuguese texts. In Proceedings of the Brazilian Symposium on Multimedia and the Web, pages 191–201, 2022.

[70] Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, , Matthew S Gerber, and Laura E Barnes. Hdltex: Hierarchical deep learning for text classification. In Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on. IEEE, 2017.

[71] Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. Hdltex: Hierarchical deep learning for text classification. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 364–371. IEEE, 2017.

[72] Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. arXiv preprint arXiv:1711.00043, 2017.

[73] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942, 2019.

[74] Loïc Lannelongue, Jason Grealey, and Michael Inouye. Green algorithms: quantifying the carbon footprint of computation. Advanced science, 2021.

[75] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100, 2023.

[76] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436–444, 2015.

[77] Guy Lev, Benjamin Klein, and Lior Wolf. In Defense of Word Embedding for Generic Text Representation, pages 35–50. Springer International Publishing, 2015.

[78] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, 2019.

[79] Enrique Leyva, Antonio González, and Raúl Pérez. Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective. Pattern Recognition, 48(4):1523–1537, 2015.

[80] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S Yu, and Lifang He. A survey on text classification: From traditional to deep learning. ACM Transactions on Intelligent Systems and Technology (TIST), 13(2):1–41, 2022.

[81] Xin Li and Dan Roth. Learning question classifiers. In Proceedings of the 19th International Conference on Computational Linguistics - Volume 1, COLING '02, page 1–7, USA, 2002. Association for Computational Linguistics.

[82] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. Neurocomputing, 461:370–403, 2021.

[83] Bing Liu. Sentence Subjectivity and Sentiment Classification, page 89–114. Studies in Natural Language Processing. Cambridge University Press, 2 edition, 2020.

[84] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. IEEE transactions on knowledge and data engineering, 35(1):857–876, 2021.

[85] 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.

[86] Zhiwei Liu, Yingtong Dou, Philip S. Yu, Yutong Deng, and Hao Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, page 1569–1572, New York, NY, USA, 2020. Association for Computing Machinery.

[87] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido M Van de Ven, et al. Avalanche: an end-to-end library for continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3600–3610, 2021.

[88] Washington Luiz, Felipe Viegas, Rafael Alencar, Fernando Mourão, Thiago Salles, Dárlinton Carvalho, Marcos Andre Gonçalves, and Leonardo Rocha. A feature-oriented sentiment rating for mobile app reviews. In Proceedings of the 2018 World Wide Web Conference, WWW '18, page 1909–1918, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[89] Zhengyi Ma, Zhicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. Pre-training for ad-hoc retrieval: hyperlink is also you need. In

Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pages 1212–1221, 2021.

[90] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. Efficient document re-ranking for transformers by precomputing term representations. In Proceedings of the 43rd International ACM SIGIR, SIGIR '20, 2020.

[91] Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. Hate speech detection: Challenges and solutions. PLOS ONE, 14(8):1–16, 08 2019.

[92] Mohamed Malhat, Mohamed El Menshawy, Hamdy Mousa, and Ashraf El Sisi. A new approach for instance selection: Algorithms, evaluation, and comparisons. Expert Systems with Applications, 149:113297, 2020.

[93] Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE transactions on pattern analysis and machine intelligence, 42(4):824–836, 2018.

[94] Karen Martins, Pedro Vaz de Melo, and Rodrygo Santos. Why do document-level polarity classifiers fail? In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1782–1794, 01 2021.

[95] Yoshitomo Matsubara, Thuy Vu, and Alessandro Moschitti. Reranking for efficient transformer-based answer selection. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, page 1577–1580, New York, NY, USA, 2020. Association for Computing Machinery.

[96] Dheeraj Mekala, Xinyang Zhang, and Jingbo Shang. Meta: Metadata-empowered weak supervision for text classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8351–8361, 2020.

[97] Luiz Felipe Mendes, Marcos André Gonçalves, Washington Cunha, Leonardo C. da Rocha, Thierson Couto Rosa, and Wellington Martins. "keep it simple, lazy" - metalazy: A new metastrategy for lazy text classification. In CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, pages 1125–1134. ACM, 2020.

[98] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: A comprehensive review. ACM Comput. Surv., 54(3), apr 2021.

[99] Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In 2020 11th international conference on information and communication systems (ICICS), pages 243–248. IEEE, 2020.

[100] Michal Moran, Tom Cohen, Yuval Ben-Zion, and Goren Gordon. Curious instance selection. Information Sciences, 608:794–808, 2022.

[101] Fernando Mourão, Leonardo C. da Rocha, Renata Braga Araújo, Thierson Couto, Marcos André Gonçalves, and Wagner Meira Jr. Understanding temporal aspects in document classification. In Marc Najork, Andrei Z. Broder, and Soumen Chakrabarti, editors, Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008, pages 159–170. ACM, 2008.

[102] Franco Maria Nardini, Cosimo Rulli, Salvatore Trani, and Rossano Venturini. Neural network compression using binarization and few full-precision weights. arXiv preprint arXiv:2306.08960, 2023.

[103] Andrew Ng. Nuts and bolts of building ai applications using deep learning. NIPS Keynote Talk, 2016.

[104] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In Proceedings of EMNLP, pages 79–86, 2002.

[105] Andrea Pasin, Washington Cunha, Marcos Goncalves, and Nicola Ferro. A quantum annealing instance selection approach for efficient and effective transformer fine-tuning. In Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval, 2024.

[106] David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. The carbon footprint of machine learning training will plateau, then shrink, 2022.

[107] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[108] Rafael B Pereira, Alexandre Plastino, Bianca Zadrozny, and Luiz HC Merschmann. Categorizing feature selection methods for multi-label classification. Artificial intelligence review, 49:57–78, 2018.

[109] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

[110] Sivaramakrishnan Rajaraman, Prasanth Ganesan, and Sameer Antani. Deep learning model calibration for improving performance in class-imbalanced medical image classification tasks. PloS one, 17(1):e0262838, 2022.

[111] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 11 2019.

[112] Julio C. S. Reis, André Correia, Fabrício Murai, Adriano Veloso, and Fabrício Benevenuto. Explainable machine learning for fake news detection. In Proceedings of the 10th ACM Conference on Web Science, WebSci '19, page 17–26, New York, NY, USA, 2019. Association for Computing Machinery.

[113] Filipe Nunes Ribeiro, Matheus Araújo, P. Gonçalves, Marcos André Gonçalves, and Fabrício Benevenuto. Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods. EPJ Data Science, 5:1–29, 2016.

[114] Bernard Rous. Major update to acm's computing classification system. Commun. ACM, 55(11):12, nov 2012.

[115] Cristòfol Rovira, Lluís Codina, Frederic Guerrero-Solé, and Carlos Lopezosa. Ranking by relevance and citation counts, a comparative study: Google scholar, microsoft academic, wos and scopus. Future Internet, 11(9), 2019.

[116] Abhinaba Roy and Erik Cambria. Soft labeling constraint for generalizing from sentiments in single domain. Knowledge-Based Systems, 245:108346, 2022.

[117] Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. How to train data-efficient llms. arXiv preprint arXiv:2402.09668, 2024.

[118] 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.

[119] Welton Santos, Washington Cunha, Celso França, Guilherme Fonseca, Sergio Canuto, Leonardo Rocha, and Marcos Gonçalves. Uma metodologia para tratamento do viés da maioria em modelos de stacking via identificação de documentos difíceis. In Anais do XXXVIII Simpósio Brasileiro de Bancos de Dados, pages 408–413. SBC, 2023.

[120] Harrisen Scells, Shengyao Zhuang, and Guido Zuccon. Reduce, reuse, recycle: Green information retrieval research. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 2825–2837, 2022.

[121] Fabrizio Sebastiani. Machine learning in automated text categorization. ACM Comput. Surv., 34(1):1–47, 2002.

[122] Burr Settles. Active Learning Literature Survey. Machine Learning, 15(2):201–221, 2010.

[123] Marco Siino, Ilenia Tinnirello, and Marco La Cascia. Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers. Information Systems, 121:102342, 2024.

[124] Naan Silva, Davi Reis, Washington Cunha, Elisa Tuler, Thiago Silva, Nicollas Silva, , and Leonardo Rocha. Integrando avaliações textuais de usuários em recomendação baseada em aprendizado por reforço. In Proceedings of the 30th Brazilian Symposium on Multimedia and the Web, 2024.

[125] Vishwanath A. Sindagi, Rajeev Yasarla, Deepak Sam Babu, R. Venkatesh Babu, and Vishal M. Patel. Learning to count in the crowd from limited labeled data. In Computer Vision – ECCV, pages 212–229, Cham, 2020.

[126] 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.

[127] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. Information Processing & Management (IP&M), 45(4):427–437, July 2009.

[128] Zhongqiang Sun, Wenhao Ying, Wenjin Zhang, and Shengrong Gong. Undersampling method based on minority class density for imbalanced data. Expert Systems with Applications, 249:123328, 2024.

[129] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pages 1165–1174, 2015.

[130] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: Extraction and mining of academic social networks. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, page 990–998, New York, NY, USA, 2008. Association for Computing Machinery.

[131] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

[132] Chih-Fong Tsai, Zong-Yao Chen, and Shih-Wen Ke. Evolutionary instance selection for text classification. J. Syst. Softw., 90(C):104–113, apr 2014.

[133] Julián Urbano, Harlley Lima, and Alan Hanjalic. Statistical significance testing in information retrieval: An empirical analysis of type i, type ii and type iii errors. In Proceedings of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pages 505–514, 2019.

[134] Felipe Viegas, Sergio Canuto, Washington Cunha, Celso França, Claudio Valiense, Leonardo Rocha, and Marcos André Gonçalves. Clusent–combining semantic expansion and de-noising for dataset-oriented sentiment analysis of short texts. In Proceedings of the 29th Brazilian Symposium on Multimedia and the Web, pages 110–118, 2023.

[135] Felipe Viegas, Sergio Canuto, Washington Cunha, Celso França, Claudio Valiense, Guilherme Fonseca, Ana Machado, Leonardo Rocha, and Marcos Gonçalves. Pipelining semantic expansion and noise filtering for sentiment analysis of short documents – clusent method. Journal on Interactive Systems, 15(1), 2024.

[136] Felipe Viegas, Sérgio Canuto, Christian Gomes, Washington Luiz, Thierson Rosa, Sabir Ribas, Leonardo Rocha, and Marcos André Gonçalves. Cluwords: Exploiting semantic word clustering representation for enhanced topic modeling. In Proceedings of WSDM '19, pages 753–761, 2019.

[137] Felipe Viegas, Washington Cunha, Christian Gomes, Antônio Pereira, Leonardo C. da Rocha, and Marcos André Gonçalves. Cluhtm - semantic hierarchical topic mod-

eling based on cluwords. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 8138–8150. Association for Computational Linguistics, 2020.

[138] Felipe Viegas, Washington Luiz, Christian Gomes, Amir Khatibi, Sérgio Canuto, Fernando Mourão, Thiago Salles, Leonardo Rocha, and Marcos André Gonçalves. Semantically-enhanced topic modeling. In Proceedings of the 27th ACM international conference on information and knowledge management, pages 893–902, 2018.

[139] Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. Fine-grained spoiler detection from large-scale review corpora. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2605–2610, Florence, Italy, July 2019. Association for Computational Linguistics.

[140] Wei Wang, Vincent W. Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. ACM Trans. Intell. Syst. Technol., 10(2), jan 2019.

[141] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Advances in Neural Information Processing Systems, 33:5776–5788, 2020.

[142] D Randall Wilson and Tony R Martinez. Reduction techniques for instance-based learning algorithms. Machine learning, 38(3):257–286, 2000.

[143] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics, pages 408–421, 1972.

[144] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS), volume 32, pages 5754–5764, 2019.

[145] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS), volume 32, pages 5754–5764, 2019.

[146] Bruna Zanotto, AP Etges, Avner Dal Bosco, EG Cortes, R Ruschel, SO Martins, AC Souza, C Valiense, F Viegas, S Canuto, et al. Pcv50 automatic classification of electronic health records for a value-based program through machine learning. Value in Health, 24:S76, 2021.

[147] Bruna Stella Zanotto, Ana Paula Beck da Silva Etges, Avner dal Bosco, Eduardo Gabriel Cortes, Renata Ruschel, Ana Claudia De Souza, Claudio M V Andrade, Felipe Viegas, Sergio Canuto, Washington Luiz, Sheila Ouriques Martins, Renata Vieira, Carisi Polanczyk, and Marcos André Gonçalves. Stroke outcome measurements from electronic medical records: Cross-sectional study on the effectiveness of neural and nonneural classifiers. JMIR Med Inform, 9(11):e29120, 2021.

[148] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. Dive into deep learning. arXiv preprint arXiv:2106.11342, 2021.

[149] Hao Zhang, Han Xu, Xin Tian, Junjun Jiang, and Jiayi Ma. Image fusion meets deep learning: A survey and perspective. Information Fusion, 76:323–336, 2021.

[150] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In Advances in Neural Information Processing Systems (NIPS), volume 28, pages 649–657. Curran Associates, Inc., 2016.

[151] Yichu Zhou and Vivek Srikumar. A closer look at how fine-tuning changes bert. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1046–1061, 2022.

Appendix A

Automatic Text Classification Datasets

Topic Classification Datasets

- **DBLP** [130] dataset is a set of research papers. This dataset is composed of 38, 128 papers in the computer science field. Each paper is classified into one of the following knowledge subareas: computer vision, computational linguistics, biomedical engineering, software engineering, graphics, data mining, security and cryptography, signal processing, robotics, and theory. As in [96], we had removed the venue for preventing a lack of information about the subarea class;

- **Books** [139] dataset is a collection of book descriptions from the Goodreads[1] website, a famous book review platform. This dataset is composed of 33, 594 book descriptions classified into the following 8 genres: children, graphic comics, paranormal fantasy, history & biography, crime & mystery thriller, poetry, romance, and young adult. As in [96], we used as data the title and description of each book;

- **ACM-Digital Library (ACM)**: a subset of the ACM Digital Library with 24, 897 documents containing articles related to Computer Science. We considered only the first level of the taxonomy adopted by ACM, composed by 11 classes.

- **20 Newsgroups (20NG)** is a classical and popular dataset for experiments in text applications of machine learning techniques. It contains 18,846 newsgroup documents[2], partitioned almost evenly across 20 different newsgroup categories.

- **OHSUMED** [63], the collection contains medical documents collected in 1991 related to 23 cardiovascular disease categories. The version we used has 18,302 documents, distributed very irregularly among the categories varying from 56 to 2876 documents per category;

- **Reuters90 (REUT)**: this is a classical text dataset, composed of news articles collected and annotated by Carnegie Group, Inc. and Reuters, Ltd. We consider here a set of 13, 327 articles, classified into 90 categories.

[1] https://www.goodreads.com/
[2] http://qwone.com/~jason/20Newsgroups/

- **Web Of Science** [70] (**WOS**) is a collection of academic articles/journals. In this work, we analyze two versions of this dataset: **WOS-5736**, with 5,736 documents classified in 11 classes; **WOS-11967**, with 11,967 documents partitioned across 33 categories;

- **4 Universities (4UNI), a.k.a, WebKB**: contains Web pages collected from Computer Science departments of four universities (Cornell (867 pages), Texas (827), Washington (1205), Wisconsin (1263) and 4,120 miscellaneous pages collected from other universities) by the Carnegie Mellon University (CMU) text learning group[3]. There is a total of 8,282 web pages, classified into 7 categories: "student", "faculty", "staff", "department", "course", "project" and "other".

- **TREC** [81], with $5,952$ documents (i.e. questions), is a question classification dataset in which the task is to classify a question into 6 main subject categories: such as human, location, entity, abbreviation, description and numeric value.

Sentiment Analysis Datasets

- **Stanford Sentiment Treebank** (**SST1**)[126] is an extension of MR with fine-grained labels ranging between very positive and very negative polarity. The SST dataset extended the MR by adding a more curated human annotation into 5 classes. **SST2** is a binary version of SST1 where only the samples with positive and negative labels were used (the samples with neutral labels wore removed).

- **pang_movie** and **vader_movie** datasets are composed of more than 10K user movie reviews. While **Movie review** (**MR**)[104] dataset is a binary (positive and negative labels) sentiment polarity composed of movie reviews[4].

- **vader_nyt** is composed of comments from the New York Times website content. Some comments are directly related to the news they were inserted to.

- **Multi-Perspective Question Answering (MPQA)**[42] is an imbalanced dataset of opinion polarity detection task. This dataset contains news documents from many sources. Each document was classified into positive or negative classes.

- **Subjectivity** dataset (**Subj**)[83], with $10,000$ documents, is a binary classification dataset in which the task is to classify a document (i.e. sentence) as subjective or objective classes.

- **yelp_reviews** [5] dataset is focused on product reviews, opinions, and comments of business.

[3] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data
[4] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[5] https://www.yelp.com/dataset/challenge

Appendix B

Automatic Text Classification Methods

SVM We use the SVM classifier with TFIDF and MetaFeatures (MFs), as it still is one of the best text classifiers capable of dealing with both high and low-dimensional representations. For the SVM implementation, we adopted the LIBSVM [24]. LIBSVM supports sequential minimum optimization that guarantees optimal Lagrange multipliers. The regularization parameter was chosen among eleven values from 2^{-5} to 2^{15} using the validation sets. Also, we used the following formulation for TFIDF weights: The TFIDF_i of element (term) i in the vector is given by $\text{TF}_i \times \text{IDF}_i$, where TF_i is the frequency of term i in the document and $\text{IDF}_i = log(\frac{N}{df_i})$, where N is the total number of documents and df_i is the document frequency of the i-th term. Note that the IDF is calculated using only the training information. In addition, we normalize the TF and IDF product result using the Euclidean norm. (a.k.a L2 normalization).

BERT [1] is an end-to-end (E2E) deep learning classifier composed of a bidirectional Transformer encoder with 24 Transformer blocks, 1024 hidden layers, and 340M parameters. The model is pre-trained with a 3.3 billion word corpus including BooksCorpus[2] (800 million words) and English Wikipedia (2.5 billion words). The original model runs on 16 TPU pods for training. In short, it predicts missing words from a sentence. The authors proposed two pre-training tasks *Masked language model (MLM)* – this technique masks words with a probability of 15% and the model is trained to predict the masked words. and (ii) *Next sentence prediction (NSP)* – it trains the model to predict whether two sentences are consecutive or not. BERT uses a multi-layer bidirectional Transformer encoder whose self-attention layer acts forward and backward. BERT redefined the state-of-the-art for 11 natural language processing tasks.

XLNet [145] is a recent E2E deep learning classifier that has outperformed BERT in 20 natural language tasks. BERT has the disadvantage of assuming that the predicted tokens are independent given unmasked tokens, a strong simplification in natural language. XLNet, a generalized autoregressive pre-training model, addresses this shortcoming by using a combination of advantages of autoregressive (AR) and autoencoder

[1] Available in https://github.com/yaserkl/
[2] Available in https://googlebooks.byu.edu/

(AE) models. In other words, the method uses a permutation language modeling objective to combine the advantages of AR and AE methods. XLNet is pre-trained with the same parameters and word corpus as BERT. In addition, three-word corpora are used, resulting in a 32.89B word corpus used to pre-training the model. Another notable difference is that training was performed with 512 TPU v3 chips.

RoBERTa [85] (***R**obustly **o**optimized **BERT** **a**pproach*) is an E2E method that, in short, includes the original BERT fine-tuning with more data and manipulations input. More specifically, it extends the study of BERT, mainly in: i) amount of data for pre-training; ii) mini-batch size; ii) training time; iii) size of the sequences considered; iv) modification of the objective task in the pre-training; and v) introduction of dynamic mask generation. The RoBERTa model was trained with an order of magnitude of data greater than the BERT. The training was carried out on a dataset in English with approximately 160GB of text containing sequences of different domains and sizes. The training was conducted on an architecture with 1024 32GB Nvidia V100 GPUs for approximately one day, meaning more computing power and time than the original BERT. The study of the Roberta model takes into account 4 objective tasks: Segment-Pair + next-sentence pretraining (NSP), Sentence-Pair + NSP, Full-Sentences, Doc-Sentences. That is, it removes the objective from the original BERT next-sentence pretraining objective. In addition to training the model, the following was increased: i) the size of the mini-batches (up to 32K sequences); ii) learning rates; and iii) the length of the sequences (up to 512). Finally, to improve the positioning of the mask proposed by BERT, the authors introduced the concept of Dynamic Masking to the model. The idea of dynamic masking is to place the MASK token in different positions in the same sequence. To avoid having to re-implement BERT, the authors duplicated each sequence 10 times ensuring that the MASK token was in different positions.

GPT-2 [109] (***G**enerative **P**retrained **T**ransformer **2***) is an OpenAI model, which advanced the state of the art in 8 NLP tasks from its GPT predecessor. The extension of the GPT2 model was performed by normalizing the layer of each sub-block, adding a normalization at the end of the self-attention layer, and training in more data. The GPT2 model was trained on a large 40GB textual corpus dataset called WebText. The model trained for the next-token prediction task and the main contribution was the application of techniques at the sub-word level (which differs from the traditional word and/or sentence level) through the Byte Pair Encoding (BPE) algorithm. Finally, the size of the models (i.e. number and size of each layer) varies between 117M parameters (small model) and 1.5B parameters (extra large model).

DistilBert [118] is a distilled extension of Bert, whose authors claim to preserve 97% of performance while keeping only half of the parameters to be learned. It is based on a technique called distillation which by definition is the approximation (through the Kulback Leiber divergence metric - KL) of a large neural network like BERT to a small

network under the assumption that the smaller one can generalize well the results of the larger one. The main modification was the reduction of the hidden layer size from 768 to 512, which resulted in a halving of the number of parameters to be learned by the network. Like Roberta, this model follows the concepts of: i) training in larger batches; ii) dynamic masking; iii) changing the objective next sentence prediction task. Finally, the DistilBERT model was trained on 8 x 16GB V100 GPUs for approximately 3.5 days under the same data as the original BERT.

DistilBert [118] is a distilled extension of Bert, whose authors claim to preserve 97% of performance while keeping only half of the parameters to be learned. Like DistilBert, **Albert** [73] (***A*** *Light* ***BERT***) is a model that aims to better train and increase BERT's performance. In this case, the authors propose techniques of parameter sharing and factorization of the embedding matrix. The parameter-sharing technique consists of sharing weights between hidden layers of the network. Parameter sharing in the attention layer is responsible for a reduction of approximately 70% of the model's parameters, which improves performance and the amount of memory needed for the model. ALBERT's second major contribution was the factorization of the Embedding matrix, where the authors proposed the use of a hidden layer matrix factorization method to reduce the number of parameters. Although it leads to a drop in model efficiency, this technique alone can reduce about 80% of the model parameters. The target task of this model was Inter Sentence Coherence Prediction, and the model was trained in a corpus containing 16 GB of uncompressed data. Finally, the size of the models varies between 12M parameters (base model) and 235M parameters (extra large model).

BART [78] (***B****idirectional and* ***A****uto-****R****egressive* ***T****ransformer*) like BERT, it is a Denoising Autoencoding model. join features like bidirectional encoder (as well as BERT) and Autoregressive (a.k.a. left-to-right) decoder (GPT2). The main idea is to include noise in the original text through an arbitrary noise function, while also learning to reconstruct the original text. Thus, in practice, a set of input tokens is replaced by the mask ([MASK] token) with noise, and the task on which the model is trained is to predict the original token for each token [MASK]. The main noise functions used are: i) token masking: adding the token [MASK] randomly as well as BERT; ii) token deletion: random token removal; iii) text infilling: pieces of text of variable size are replaced by a single token [MASK]; iv) sentence permutation: randomization of sequences based on a pre-defined period; and v) Document Rotation: after choosing a random token, the sequence is rotated.

All E2E methods were implemented with support from the PyTorch-based transformers library[3]. To guarantee the reproducibility and sharing of code, we make the implementations used in this work available on GitHub[4].

[3]Available in https://huggingface.co/transformers/
[4]Available in https://github.com/waashk/instanceselection/

Appendix C

Alternatives for the IS Input Representation

There exist several options to use as (vectorial) representation input for the IS methods, which include the TF-IDF weighting-scheme-based representation, static embeddings (Word2vec, GloVe, fastText), and contextual embeddings, resulting from fine-tuning a model or using a pre-trained model without any tuning, aka, the zero-shot approach. As previously discussed, we have decided to adopt the TF-IDF weighting-scheme-based representation due to several factors, including simplicity, popularity, effectiveness, and understandability. Next, we present results that strengthen this decision.

Static Embeddings A first alternative for replacing TFIDF in our experiments would be static embeddings, such as FastText[68]. A common strategy to represent documents with FastText is using the representation derived from averaging the respective embedding vectors of the words present in the document. Adopting this document representation, Cunha et al. [31] demonstrated that the deployment of classification techniques based on static embeddings results in considerable effectiveness losses when compared to standard TFIDF (+SVM) in several of the exploited datasets. Additionally, exploring embeddings can considerably increase the computational costs – 1.5x and 31.1x slower than TF-IDF.

Contextual Embeddings via Model Fine-Tuning Another possibility would be using contextual embeddings built by modern neural Transformer architectures, as they leverage most state-of-the-art models in many natural language processing tasks through (i) a model fine-tuning or (ii) pre-trained model without any tuning, aka, zero-shot approach. In both cases, the Transformers architecture would act as a *textual Encoder* that represents the input raw text into a highly dimensional (e.g., 768-dimensions) vector space.

The first alternative (i) performs the fine-tuning process in each specific context (dataset) to create the text encoder, which is then applied to the documents present in each dataset to obtain the respective contextual embedding vectors. Despite the potential benefits in terms of effectiveness [151], in the context of IS, it is unfeasible to tune before

selecting, given the high cost of this procedure. In other words, it does not make any sense to perform fine-tuning as a pre-processing step and, after that, select the most representative instances and finally train the (same or other strong) classification model again.

Contextual Embeddings via Zero-Shot Approach The zero-shot approach is a less complex and computationally expensive option for contextual embeddings since we only need to load a pre-trained model and represent each document in a vector of 768 dimensions without the cost of any tuning. In this representation, we use a token included in the model representing the sentence (in our case, the whole document), known as "CLS". In practice, this token corresponds to a pooling calculated with self-attention (instead of average or max-pooling) of the vectors of the tokens present in the document [148]. We will investigate other zero-shot contextual embedding alternatives for future work, such as MUSE [72] and SentenceBERT [111].

Next, we present the results obtained using the BERT transformer model as a pre-trained textual encoder in a zero-shot approach to generate the input representation for the five best IS methods (Table 3.8). In Table C.1, we present the time increase rate (Fold Average of a 10-Fold CV procedure) achieved by each selection method regarding only the selection time. The time increase rate is calculated by dividing the selection time using contextual embeddings and the selection time using TF-IDF as representation input, respectively. A red color scale for each line (dataset), accompanied by the respective value, is shown in the Table. The darker a cell, the more computationally expensive the corresponding method is for the corresponding dataset.

As we can see, the use of contextual embeddings as IS input is more costly when compared to the TFIDF input (dark red cells) with the IS methods: **CNN**, **EGDIS** and **CIS**. In more detail, considering the contextual embedding representation, the **CNN** method – considered the best method evaluated in this work in terms of the constraints' tripod – becomes computationally much more expensive. On average, there is a time increase of 9.8x (ranging from 1.47x – yelp_reviews to 43.09x – SST1). Indeed, using contextual embeddings with CNN to select instances becomes 1.3x to 3.0x more expensive than training the model with all data (NoSel) in datasets such as DBLP, Books, ACM, 20NG, OHSUMED, and SST1.[1]

Significant increases in computational costs are also observed for **EGDIS**. Previously, considering TFIDF as input, this method had an average speed-up of 2x. On the other hand, the use of contextual embedding representation made this method, on average, 25.11x more expensive (varying between 7.13x to 66.39x). Both CNN and EGDIS use the KNN model iteratively in their approaches. With highly-dimensional, high-density representations, as in the case of contextual embeddings, all (768) dimensions of the

[1]Due to space limitations, we provide an online table containing the exact times (in seconds) of the application of all methods and analyzes presented here in the following link: https://shorturl.at/zCLW7

representation must be considered during the distance calculation step, making this calculation very expensive. On the other hand, the calculation of the distances with sparse representations (such as TFIDF) considers only the non-zero dimensions, thus benefiting from efficient data structure implementations such as inverted indexes that accelerate the calculation enormously.

task	dataset	CNN	LSSm	LSBo	EGDIS	CIS *
Topic	DBLP	6.31	0.94	0.81	9.80	75858.75
	Books	10.70	1.00	0.75	8.89	135162.89
	ACM	12.70	0.90	0.78	21.04	146282.63
	20NG	8.54	0.95	0.65	7.13	153032.52
	OHSUMED	7.94	0.94	0.76	9.60	27792.44
	Reuters90	3.59	0.86	0.82	13.00	27829.52
	WOS-11967	4.29	0.92	0.75	9.58	17793.90
	WebKB	2.14	0.95	0.90	7.79	5695.73
	TREC	13.37	1.53	0.90	33.30	929.55
	WOS-5736	2.76	0.91	0.77	8.00	4140.33
Sentiment	SST1	43.09	1.43	0.69	49.84	9034.48
	pang_movie	7.07	1.01	0.97	33.36	13742.53
	MR	19.87	1.55	0.84	48.13	6882.55
	vader_movie	6.45	0.98	0.94	33.06	12958.81
	MPQA	11.52	1.47	0.90	66.39	1569.90
	Subj	5.12	1.50	1.11	44.80	7822.66
	SST2	16.87	1.51	0.80	45.80	5995.09
	yelp_reviews	1.47	0.94	0.88	7.78	4865.74
	vader_nyt	2.39	0.86	0.17	19.83	3925.68

Table C.1: Selection time increase rate: Ratio between the selection time using contextual embeddings and the selection time using TF-IDF as representation input, respectively.

Last, the **CIS** method was already expensive when using TFIDF as input representation. Following this trend, the contextual representation made the selection process even more expensive. This method uses a weak model (KMeans) to estimate the impact of the chosen instances in each iteration. Therefore, it presents the same cost problem mentioned above when considering iterative pairwise distance calculations applied to a dense representation. CIS is an iterative method in which the number of iterations is defined through a formula that considers the dataset's characteristics (such as the number of instances and features). Therefore, we were able to run only three iterations for each dataset to estimate the time needed to run for all iterations. Since the time variation between iterations is low in both representations, considering only three iterations, this time estimation is reasonable. According to our estimates, it would take over 200 years to run entirely for all datasets.

CNN, EGDIS, and CIS break the efficiency pillar of the tripod. LSSm and LSBo strategies, on the contrary, have compatible selection times when compared to the TFIDF input. LSSm is, on average, 11% more expensive while LSBo, becomes, on average, 20% faster in the selection phase using zero-shot contextual representations. Both methods are based on local sets. As such, they could be viable alternatives to use along with zero-shot contextual embeddings. Next, we analyze the effectiveness of both IS methods when using this representation.

Task	dataset	LSSm		LSBo	
		TFIDF	Contextual	TFIDF	Contextual
Topic	DBLP	**81.1(0.8)** ▲	79.8(0.7) ▼	**79.1(0.6)** ▲	78.8(0.8) ▼
	Books	**88.8(0.5)** ▲	85.9(0.4) ▼	**84.0(0.5)** •	**84.5(0.4)** •
	ACM	**69.6(1.3)** •	**68.5(1.1)** •	**63.8(1.5)** •	**65.3(1.3)** •
	20NG	**90.7(0.5)** ▲	83.6(0.7) ▼	**90.7(0.6)** ▲	82.1(0.6) ▼
	OHSUMED	**73.8(0.5)** ▲	71.1(1.2) ▼	**68.8(1.2)** ▲	63.9(15.8) ▼
	Reuters90	**38.1(1.7)** ▲	36.9(1.7) ▼	**36.5(2.2)** •	**38.6(2.7)** •
	WOS-11967	**86.4(0.9)** ▲	85.6(0.6) ▼	**84.9(0.6)** •	**85.3(0.5)** •
	WebKB	**80.6(1.8)** ▲	78.2(1.9) ▼	**76.2(2.1)** •	**78.0(2.4)** •
	TREC	**95.0(0.7)** ▲	92.8(1.3) ▼	**95.0(1.1)** ▲	92.1(1.1) ▼
	WOS-5736	**88.0(1.1)** •	**88.6(1.0)** •	**86.5(1.4)** •	**86.8(1.2)** •
Sentiment	SST1	**53.4(0.9)** •	**53.0(0.9)** •	**53.2(0.9)** •	**52.2(0.7)** •
	pang_movie	**88.5(0.5)** •	**88.2(0.6)** •	**88.0(0.6)** ▲	87.4(0.6) ▼
	MR	**89.0(0.6)** •	**88.6(0.5)** •	39.3(12.3) ▼	**87.6(0.5)** ▲
	vader_movie	**90.8(0.7)** •	**91.0(0.6)** •	**90.5(0.4)** •	**90.0(0.7)** •
	MPQA	**90.0(0.7)** ▲	89.0(0.7) ▼	**89.9(0.6)** ▲	89.4(0.5) ▼
	Subj	**95.4(0.7)** •	**96.7(0.4)** •	**95.6(0.5)** •	**96.0(0.3)** •
	SST2	**92.9(0.5)** •	**92.9(0.5)** •	**93.0(0.7)** •	**92.1(0.7)** •
	yelp_reviews	**97.7(0.3)** ▲	97.3(0.3) ▼	**97.4(0.3)** •	**97.4(0.5)** •
	vader_nyt	**83.9(0.9)** •	**83.3(0.9)** •	**83.6(1.2)** •	**83.1(0.8)** •

Table C.2: Effectiveness Analysis. Statistical comparison between the TFIDF and Contextual embeddings used as input of the **LSSm** and **LSBo** approaches and applied to the best classifier per dataset (Table 3.5). Legend: (a) ▲: the IS method with the specific input (TFIDF or Contextual) is statistically superior to its pair; (b) •: the IS method with the specific input statistically equivalent to its pair; (c) ▼: the IS method with the specific input statistically worse than its pair.

Table C.2 experiment consists of comparing the use of TFIDF and contextual embeddings as input for the LSSm and LSBo approaches applied to the best classifier per dataset (Table 3.5). The experiments were executed using a 10-fold cross-validation procedure. To compare the average results on our cross-validation experiments, we assess the statistical significance employing the paired t-test with 95% confidence, which, in this case, is presumably resilient to any breaches of the normality assumption and is strongly advised above signed-rank tests for hypothesis testing on mean effectiveness [133, 65].

The use of contextual embeddings applied as input to the LSSm strategy caused statistically significant losses in 10 datasets, tying in the remaining ones. More specifically, this method worsened the results in eight topic datasets (out of the ten analyzed) and two sentiment datasets (MPQA and yelp_reviews) and it did not improve the results in any dataset. On a smaller scale, the LSBo also looses effectiveness – it becomes statistically worse on six datasets, tying in 11 and improving over TFIDF only in one (the MR dataset).

In summary, directly using contextual embeddings as input for IS methods have been demonstrated to be inefficient or ineffective. Overall, the above exercise revealed that using contextual embeddings along with IS methods is not trivial and will require further research considering the current SOTA of both fields.

Appendix D

Average Total Time for model training

In this section, we present the cost of each method in terms of the model construction time. This full training process comprises the total times for the preprocessing stages (including the IS step) and ML training model, i.e., the time to fine-tune the transformer-based ATC model. The metric is the overall time in seconds, average by the number of folds: the smaller datasets were executed using k=10-fold partition, while for the larger ones, we adopted 5 folds due to the cost of the procedure.

task	dataset	NoSel	biO-IS	E2SC (LR)	CNN	LSSm	LSBo	EGDIS	CIS	IB3
Topic	DBLP	4,988.12	2,814.34	3,797.15	4,279.68	5,741.91	4,067.09	2,569.37	43,934.76	6,712.28
	Books	4,412.46	3,108.34	3,240.60	4,184.31	5,003.93	3,630.62	2,157.75	15,527.37	6,921.55
	ACM	3,050.29	2,075.06	2,396.34	2,059.48	3,043.58	2,221.29	1,612.21	6,294.58	2,589.20
	20NG	2,781.84	1,952.26	2,147.62	2,451.39	3,189.84	2,795.54	1,224.83	2,810.80	3,781.91
	OHSUMED	2,780.05	1,751.89	2,052.76	1,871.81	2,617.93	1,468.04	1,763.75	7,053.94	2,014.42
	Reuters90	2,156.85	1,365.51	1,664.12	1,472.28	2,112.80	1,037.59	1,194.61	2,645.52	2,893.43
	WOS-11967	1,759.63	715.52	871.62	1,288.53	1,778.72	930.67	841.51	2,447.60	1,012.73
	WebKB	602.47	451.06	519.27	546.35	592.25	330.29	404.61	1,177.29	506.99
	Twitter	513.81	358.36	380.17	299.62	502.12	282.88	251.01	1,278.21	318.17
	TREC	463.49	308.39	376.77	357.78	414.21	373.32	354.68	2,191.12	375.71
	WOS-5736	820.21	410.19	457.37	533.08	749.85	356.79	395.16	615.66	461.57
Sentiment	SST1	809.63	394.38	626.19	661.14	850.87	969.46	667.06	3,880.55	904.65
	pang_movie	681.27	442.63	530.34	456.67	651.43	433.68	320.28	1,283.10	439.79
	MR	672.37	478.46	531.10	562.97	734.18	614.14	331.23	2,399.94	439.26
	vader_movie	675.44	463.00	532.20	424.94	621.96	437.97	317.98	1,262.84	757.01
	MPQA	676.64	515.57	575.24	310.87	788.79	507.07	424.02	9,427.23	798.48
	Subj	708.45	534.87	536.94	394.77	660.56	357.19	238.90	1,801.34	308.33
	SST2	619.91	435.52	499.81	425.50	711.50	767.42	280.23	1,981.80	345.13
	yelp_reviews	760.71	328.94	349.82	364.26	660.19	330.47	242.84	525.04	267.85
Large	AGNews	18,008.45	8,902.08	10,269.45	-	-	-	-	-	-
	yelp_2013	37,274.03	19,016.57	25,651.86	-	-	-	-	-	-
	MEDLINE	122,030.00	63,660.48	76,321.75	-	-	-	-	-	-

Table D.1: Average Total Time for model training.

Remarks: For this Ph.D. dissertation, considering all considered IS methods, classifiers, and variations, we run **four thousand** experiments using SOTA Transformers corresponding to about **5,600** hours (233 days) of experiments.

Appendix E

Wrongly Predicted Instances Potential

Table E.1 presents the number (and percentage) of wrongly predicted instances by both KNN and LR weak classifiers for each dataset of our experimental setup.

task	dataset	# Training Instances	KNN (% error)	LR (% error)
Topic	DBLP	34315	7179 (20.9%)	6932 (20.2%)
	Books	30234	6532 (21.6%)	5942 (19.7%)
	ACM	22402	6247 (27.9%)	5562 (24.8%)
	20NG	16954	2658 (15.7%)	2400 (14.2%)
	OHSUMED	16471	4699 (28.5%)	5006 (30.4%)
	Reuters90	11977	3855 (32.2%)	3659 (30.6%)
	WOS-11967	10770	2736 (25.4%)	1724 (16.0%)
	WebKB	7376	2497 (33.9%)	1753 (23.8%)
	Twitter	6297	1561 (24.8%)	1516 (24.1%)
	TREC	5356	2310 (43.1%)	1825 (34.1%)
	WOS-5736	5162	1166 (22.6%)	596 (11.5%)
Sentiment	SST1	10669	7258 (68.0%)	6456 (60.5%)
	pang_movie	9594	2605 (27.2%)	2277 (23.7%)
	MR	9595	3000 (31.3%)	2383 (24.8%)
	vader_movie	9510	2501 (26.3%)	2167 (22.8%)
	MPQA	9545	6078 (63.7%)	1862 (19.5%)
	Subj	9000	1413 (15.7%)	1031 (11.5%)
	SST2	8651	2466 (28.5%)	1865 (21.6%)
	yelp_reviews	4500	735 (16.3%)	307 (6.8%)
Large	AGNews	102080	9851 (9.7%)	8742 (8.6%)
	yelp_2013	268014	143944 (53.7%)	106594 (39.8%)
	MEDLINE	688337	150832 (21.9%)	91724 (13.3%)
	average	-	30.0%	22.8%

Table E.1: Number of Wrongly Predicted Instances Potential (percentual error)

These results indicate that E2SC's weak-classifier (KNN) predicts around 30% of the instances as "hard to classify". This sets up an upper limit on the method's reduction capability to around 70% – a reminder that "hard-to-classify" instances are never removed from the dataset. When using the LR weak classifier, this number decreases to an average of 22.8%, increasing the potential of the redundancy-based removal. Furthermore, our experiments have also shown that a significant portion of these "hard to classify" instances can be removed without affecting the model's effectiveness, further improving its training efficiency.

Appendix F

Impact of Noise Insertion and Removal

This experiment is similar to the one performed in section 5.1.1, where we artificially inserted noise into the datasets. The main difference is that, in here, we also measure the effectiveness of RoBERTa when noise is inserted and its effectiveness after applying the entropy-based approach proposed for noise removal. The main idea of this experiment is to measure the effectiveness of the transformer-based model when artificially introducing potentially noisy documents and subsequently removing them using the entropy-based step of our method.

Accordingly, for each of the datasets, we randomly switched the label (real class) of a fixed percentage pair of documents, producing a total of 5% and 10% of noisy instances with switched labels in each dataset. The obtained results are presented in Table F.1.

		Without manually inserted noise	5% Noise		10% Noise	
task	dataset	NoSel	NoSel	bio-IS (only entropy-based approach)	NoSel	bio-IS (only entropy-based approach)
Topic	DBLP	81.4(0.5)	80.1(0.7)	79.9(0.9)	79.2(0.7)	79.0(0.8)
	Books	87.2(0.6)	85.5(0.7)	85.3(0.6)	83.8(0.7)	84.1(0.6)
	ACM	70.3(1.4)	68.2(1.5)	68.0(1.4)	68.2(1.2)	67.0(1.2)
	20NG	86.0(0.7)	85.3(0.8)	85.4(0.6)	83.9(0.8)	84.4(0.9)
	OHSUMED	77.8(1.2)	75.6(0.9)	74.4(1.1)	74.5(1.0)	73.3(1.0)
	Reuters90	41.9(2.2)	42.1(2.3)	41.5(2.4)	42.1(2.0)	41.4(2.4)
	WOS-11967	86.8(0.4)	86.2(0.7)	86.7(0.4)	85.4(0.7)	86.2(0.6)
	WebKB	83.0(2.0)	80.5(1.8)	80.9(1.4)	79.5(1.6)	79.1(1.8)
	Twitter	78.4(1.8)	77.7(1.8)	74.2(1.7)	75.5(2.3)	74.2(2.2)
	TREC	95.5(0.5)	94.2(0.8)	93.8(0.9)	93.1(1.2)	92.5(1.3)
	WOS-5736	90.5(0.9)	89.8(1.0)	89.9(0.9)	89.0(0.7)	88.8(1.0)
Sentiment	SST1	53.8(1.3)	52.9(1.8)	52.7(1.1)	52.7(1.1)	51.2(1.3)
	pang_movie	89.0(0.4)	87.6(0.7)	87.3(0.5)	86.6(0.7)	86.5(0.6)
	MR	89.0(0.7)	87.8(0.5)	87.9(0.5)	86.3(0.5)	86.7(0.6)
	vader_movie	91.3(0.5)	90.3(0.5)	90.1(0.7)	89.1(0.9)	88.8(0.6)
	MPQA	90.2(0.8)	89.4(0.6)	87.5(0.5)	89.0(0.5)	88.6(1.0)
	Subj	96.9(0.4)	96.0(0.4)	95.7(0.6)	94.5(0.9)	94.9(0.6)
	SST2	93.2(0.6)	91.6(0.7)	91.4(0.7)	90.8(0.7)	90.9(0.5)
	yelp_reviews	97.9(0.4)	96.8(0.7)	97.3(0.5)	95.6(1.4)	96.7(0.5)

Table F.1: Impact of Noise Insertion and subsequent Removal.

This experiment demonstrates that noisy (training) instances have the potential to reduce the effectiveness of the model by introducing misleading patterns, Indeed, according to Table F.1 results, comparing the columns NoSel (Without manually inserted noise) and NoSel (5% and 10% Noise), it is possible to notice that manually adding noise in the datasets slightly degraded the models (varying between 0.6% up to 3.0%; and from 1.3% to 4.2%, for 5% and 10% of manually inserted noise). Notice that the degradation in effectiveness is smaller than the amount of noise inserted, pointing out a certain resilience of the Transformer to the inserted noise.

On the other hand, even though our proposal was able to remove a significant portion of these noise instances as reported in the main experiments of the article, there was almost no impact in terms of effectiveness (neither positive nor negative).

Put together, these results provide compelling evidence that, for the sake of effectiveness, transformer-based models are very resilient to both **noise insertion** and **noise removal**. As our method does not perform data (label) correction, we hypothesize that potential gains coming from cleaner patterns are counter-balanced by less data to learn. Therefore, the most observable impact of removing potentially noisy instances falls on efficiency, which demonstrated significant improvements in our main experiments when disregarding these instances during training time.

Appendix G

Weak-classifier algorithms' hiperparameterization

The implementations of Logistic Regression, Linear SVM, Random Forest, Decision Trees, Naive Bayes, and Nearest Centroid are from scikit-learn. XGBoost and lightGBM are from the respective authors' implementation-based packages. For KNN, we adopted an approximated solution (HNSW [93]), which is effective, computationally cheap, and scalable.

Table G.1 presents the hiperparameterization for each algorithm. These parameters were set based on the best values according to our empirical preliminary experimentation. Omitted parameters are the library's default.

method	parameters
Logistic Regression	{'C': 1.0, 'penalty': 'l2', 'dual': False, 'tol': 0.0001, 'fit_intercept': True, 'intercept_scaling': 1, 'solver': 'warn', 'max_iter': 1000, 'multi_class': 'warn', 'warm_start': False, 'n_jobs': -1}
Linear SVM	{'C': 1.0, 'intercept_scaling': 1, 'fit_intercept': True, 'max_iter': 1000, 'penalty': 'l2', 'multi_class': 'ovr', 'dual': False, 'tol': 0.001, 'class_weight': None}
Random Forest	{'n_estimators': 200, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'criterion': 'gini', 'max_features': 'auto','warm_start': False, 'oob_score': False, 'bootstrap': True}
Decision Trees	{'criterion': 'gini', 'splitter': 'best', 'min_samples_split':2, 'min_samples_leaf':1, 'min_weight_fraction_leaf':0.0, 'min_impurity_decrease':0.0, 'presort':False}
Naive Bayes	{'alpha': 1.0, 'fit_prior': True, 'class_prior': None}
XGBoost	{'objective': 'binary:logistic', 'eval_metric': 'logloss', 'learning_rate': 0.1, 'max_depth': 5, 'subsample': 0.1,'tree_method': 'auto', 'n_estimators': 100}
lightGBM	{'boosting_type': 'gbdt', 'colsample_bytree': 1.0, 'importance_type': 'split', 'learning_rate': 0.1, 'max_depth': -1, 'min_child_samples': 20, 'min_child_weight': 0.001, 'min_split_gain': 0.0, 'n_estimators': 100, 'num_leaves': 31, 'reg_alpha': 0.0, 'reg_lambda': 0.0, 'subsample': 1.0, 'subsample_for_bin': 200000, 'subsample_freq': 0}
KNN	{'n_neighbors': 10, 'weights': 'uniform', 'algorithm': 'auto', 'leaf_size': 30, 'metric': 'euclidean'}
Nearest Centroid	{'metric': 'euclidean'}

Table G.1: Weak-classifier algorithms' hiperparameterization