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

Washington Cunha
Federal University of Minas Gerais
Brazil
washingtoncunha@dcc.ufmg.br

Celso França
Federal University of Minas Gerais
Brazil
celsofranca@dcc.ufmg.br

Guilherme Fonseca
Federal University of São João del Rei
Brazil
guilhermefonseca8426@aluno.ufsj.edu.br

Leonardo Rocha
Federal University of São João del Rei
Brazil
lcrocha@ufsj.edu.br

Marcos André Gonçalves
Federal University of Minas Gerais
Brazil
mgoncalv@dcc.ufmg.br

ABSTRACT

Transformer-based deep learning is currently the state-of-the-art in many NLP and IR tasks. However, fine-tuning such Transformers for specific tasks, especially in scenarios of ever-expanding volumes of data with constant re-training requirements and budget constraints, is costly (computationally and financially) and energy-consuming. In this paper, we focus on **Instance Selection (IS)** – a set of methods focused on selecting the most representative documents for training, aimed at maintaining (or improving) classification effectiveness while reducing total time for training (or fine-tuning). We propose **E2SC-IS** – **E**ffective, **E**fficient, and **S**calable **C**onfidence-Based **IS** – a two-step framework with a particular focus on Transformers and large datasets. E2SC-IS estimates the probability of each instance being removed from the training set based on scalable, fast, and calibrated weak classifiers. E2SC-IS also exploits iterative heuristics to estimate a near-optimal reduction rate. Our solution can reduce the training sets by 29% on average while maintaining the effectiveness in **all** datasets, with speedup gains up to 70%, scaling for very large datasets (something that the baselines cannot do).

CCS CONCEPTS

• **Information systems** → **Document filtering**.

KEYWORDS

Instance Selection, Transformer-Based Text Classification

ACM Reference Format:
Washington Cunha, Celso França, Guilherme Fonseca, Leonardo Rocha, and Marcos André Gonçalves. 2023. 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), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591638

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591638

1 INTRODUCTION

Automatic Text Classification (ATC) approaches can be employed as helpful tools, allowing for the analysis and organization of large amounts of data. ATC has had increasing importance in challenging scenarios, including relevance feedback [19], sentiment analysis [6], product reviews [30], 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 [18]. The costs of obtaining large amounts of labeled data can also be ameliorated by approaches such as crowd [42] and soft labeling [39].

ATC has recently experienced tremendous advances, mainly due to the introduction of Transformer-based deep learning approaches, considered the current state-of-the-art (**SOTA**) [15]. These approaches can be divided into two steps: (i) pre-training; and (ii) domain transfer. The pre-train step involves learning the model weights employing an unsupervised task (e.g., Next Sentence Prediction [15]). The fine-tuning step is supervised, applied to a domain-specific labeled dataset and allows for further model optimization. According to Andrew Ng [37], 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 [4], 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.

Despite faster, fine-tuning is still a costly process that demands expensive computational resources in terms of computational power and memory demands. For instance, 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, there are several scenarios in which adopting fine-tuned deep-learning approaches can be challenging (if not impractical) despite potential effectiveness gains. For instance, consider a textual classifier applied in a scenario that requires continuous re-training (e.g., fraud detection[29], product tagging[2], and recommendation[9]). 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, delaying the learning of the temporal changes in the input data, which, in turn, can affect the effectiveness of the task[36].

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 paper, we run **four thousand** experiments using SOTA Transformers corresponding to about **4,200** hours (175 days) of experiments. Any reductions could bring benefits from several perspectives (financial, energy, etc.).

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, efficient, 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 [10]. In particular, we focus on **Instance Selection (IS)** techniques. In contrast to traditional Feature Selection approaches, in which the main objective is to select the most informative terms, **IS** methods are focused on selecting the most representative instances for the training set [17]. The intuition behind IS 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. IS methods should simultaneously guarantee the following constraints tripod: (1) training set reduction; (2) high effectiveness; and (3) high efficiency. In sum, the main objective of Instance Selection methods is *to maintain (or even improve) classification model effectiveness through an additional preprocessing step while reducing total time – in the case to train (or fine-tune) the Transformer models.*

Despite its potential, the application of IS methods for ATC, especially in the deep learning scenario, has been under-investigated. Indeed, most of the IS methods have been proposed and studied only on small tabular datasets and the selected instances were applied as input only to weak classifiers (e.g. KNN). In contrast, the datasets in ATC are unstructured, larger, and more complex. As deep learning transformers 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.

As far as we know, [12] was the first work that extensively studied the behavior of these IS techniques in the context of transformer approaches (such as BERT, RoBERTa, BART, and GPT) in the ATC field. The authors established that IS methods were able to reduce the training set by up to 90% while maintaining effectiveness, with total time speedups between 1.04 and 2.24 times. However, the best-considered instance selection methods were able to respect the "tripod" restrictions (effectiveness x reduction x total cost) in just about half of the tested datasets. In the remaining datasets, especially the large ones, the use of IS approaches caused an overhead in terms of the total time to generate the model. Therefore, despite the potential, there is much room for developing new IS methods focused on transformer-based architectures and large [1] datasets.

[1]We consider large-scale datasets for ATC, those with more than 100K documents [47].

In this context, the main contribution of this paper is the proposal of **E2SC-IS** – **E**ffective, **E**fficient, and **S**calable **C**onfidence-based **I**nstance **S**election – a novel two-step framework[2] aimed at large datasets with a special focus on transformer-based architectures. E2SC-IS 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 1.

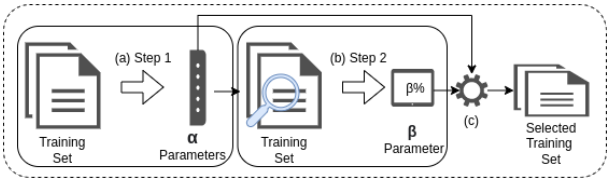

Figure 1: The proposed E2SC-IS Framework.

E2SC-IS´s **first** step (Fig. 1a) aims to assign a probability to each instance being removed from the training set (α). We adopt an exact KNN model to estimate the probability of removing instances, as it is considered a **calibrated**[3] and computationally cheap model [8]. 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 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 1(c)), β% instances are randomly sampled, weighted by the α distribution, to be removed from the training set.

The specific research questions (**RQ's**) we aim to answer are:

RQ1. *Is **E2SC-IS** capable of reducing the training set while keeping classifier effectiveness for each investigated scenario (dataset)? How does **E2SC-IS** compare with other SOTA IS approaches regarding this tradeoff?* RQ1´s goal is to investigate and compare our proposal with SOTA IS baselines regarding the tradeoff between the first two constraints of the "tripod": effectiveness and training set reduction.
RQ2. *What is the impact of applying **E2SC-IS** in the text classification models' total construction time?* We propose assessing the effect of the application of the IS approaches investigated in RQ1 regarding the third pillar of the tripod in the context of potential time reductions for the full process, which comprises the total times for preprocessing stages (including IS step) and ML training

[2]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/
[3]A calibrated classifier[38] 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.

model. Particularly, regarding the ML training model, we focus on the time to train the **Transformers** used as classifiers.

RQ3. *How flexible is the **E2SC-IS** framework to adjust to different application/task requirements?* As mentioned, the traditional IS strategies do not scale for the big data scenario (e.g., datasets with more than 100K instances). RQ3´s main objective is to demonstrate the flexibility of **E2SC-IS** by showing how its steps can be modified to accommodate different requirements posed by distinct scenarios, mainly those associated with big data.

In our experimental evaluation, we compare **E2SC-IS** with **six** robust state-of-the-art instance selection baseline methods considering as input of the best of **six** deep learning text classification methods in a large benchmark with **19** datasets. Our solution managed to significantly reduce the training sets (by **27%** on average; maximum of 60%) while maintaining the same levels of effectiveness in **18** datasets (RQ1), with speedups of **1.25** on average (RQ2) (maximum of 2.04). To demonstrate the flexibility (RQ3) of our framework to cope with large datasets, we propose two modifications. The first one replaces the interactive strategy to optimize the parameter β with a heuristic based on extracting statistical characteristics of the input dataset. The second modification replaces the exact KNN with an approximate solution with logarithmic complexity, allowing a more scalable and efficient search for the nearest neighbors.

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 (RQ1), with speedups of **1.37** on average (RQ2). In addition, the framework scaled to the large datasets (RQ3), reducing them by up to 40% while statistically maintaining the same effectiveness with speedups of **1.70x**.

2 RELATED WORK

The literature categorizes IS methods according to the adopted paradigm [12]: condensation [20], editing [45], hybrid [26], density [31], clustering based [35] and spatial hyperplane [7]. Briefly, condensation algorithms remove noise by creating subsets, which later reduce the number of instances. In contrast, editing algorithms remove noisy instances employing filters. Hybrid algorithms attempt to combine condensation and editing paradigms. Density-based approaches try to keep those present in denser regions. Similarly, clustering-based strategies first perform a method to aggregate instances and then select instances from each cluster. Last, Spatial HyperPlane algorithms divide the hyperplane space of the features by choosing representative instances of each subspace later. As we will detail below, **E2SC-IS** can be classified as a hybrid IS method.

In [12], the authors compared the most traditional and recent IS approaches applying them in the ATC context. None of the analyzed approaches respected the "tripod" restrictions (effectiveness vs. reduction vs. total cost) for all tested datasets. Despite that, we selected the six best IS methods as baselines to compare with our approach. In the following, we detail all the baseline IS methods.

The Condensed Nearest Neighbor (**CNN**) [20], a popular method within the condensation category, starts from a solution set S containing a random instance of each class. It iteratively predicts the class for each instance x in the original set of instances T, including 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. The time complexity of CNN is $O(n^3)$, where n is the size of the original set.

In [26], the authors proposed Local Set-based Smoother (**LSSm**), an editing approach, and Local Set Border Selector (**LSBo**), a hybrid approach. Both methods leverage the concept of local set (LS), a set of instances 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)).

In LSSM the set S is composed of instances that have usefulness higher than harmfulness ($u(e) > h(e)$), both concepts formally defined in the respective work. An instance e with high usefulness has importance/influence for many other instances and thus should 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 $O(n^2)$.

Other representative hybrid algorithms include IB1, IB2, IB3 [1] methods. 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. IB3 achieved the best results compared to the other two and will be used as one of our baselines. The time complexity of **IB3** algorithm is $O(n^2 \log(n))$.

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 effectiveness improvements. As these algorithms have only a local view of the dataset, both reduction and effectiveness can be limited to the algorithm knowledge of the specific class. To address these limitations, the authors in [31] propose two global density-based IS algorithms called global density-based instance selection (GDIS) and enhanced global density-based instance selection (**EGDIS**). GDIS 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 tabular data, GDIS achieves good classification accuracy values but with a decrease in reduction rate. EGDIS addresses this issue using an *irrelevance function* that determines the number of neighbors from another class. Since EGDIS presents the best trade-off between reduction and accuracy, with an $O(n^2)$ complexity, we will focus on it.

Finally, the Curious Instance Selection (**CIS**) [35] 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. The algorithm's output is a matrix representing the trade-off between model improvement and the selected data size. The time complexity of the CIS method is $O(n^3)$.

Similarly to IB3, which also belongs to the hybrid category, **E2SC-IS** 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 IB3, **E2SC-IS** 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 state-of-the-art, as our experiments shall demonstrate.

3 THE PROPOSED FRAMEWORK: E2SC-IS

Given a set of instances $X = \{x_1, x_2, ..., x_M\}$, the proposed **E2SC-IS** framework consists of two main steps. The **first** step (Figure 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 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 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 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.

3.1 Fitting α Parameters

E2SC-IS first step assigns a probability to each instance being removed from the training set ($\alpha(x)$). The **first hypothesis (H1)** of **E2SC-IS** 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-IS instantiation, we adopt as f the brute-force (exact search) k-nearest neighbor (KNN) model to estimate the probability of removing instances. In Section 3.1.1, 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 4.6.1 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 3.1.1, 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 3.2), 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 ($P_R(x) = 0$), 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.

3.1.1 ***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** [38]. For example, in instances predicted with 80% confidence, a calibrated classifier is correct in roughly 80% of

[4]For those classifiers, we adopted the same procedures and hyperparameters as in [13].

the cases. As our proposed framework removes instances based on prediction confidence, it is of paramount importance that the adopted classifier be calibrated. In [8], the authors provided graphical evidence that KNN is indeed a calibrated classifier. To confirm this result, we analyze the behavior of the weak classifiers using the Brier Score (BS) [3], a scoring rule applied to measure the accuracy of probabilistic predictions, thus, a proper metric to estimate the model calibration. According to [3], $BS = \frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C}(P(Y = y_c|x_i) - o_{ci})^2$, where o_{ci} is 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?* Achieving these three (potentially conflicting) requirements – calibration, efficiency, and effectiveness – at the same time is hard, so we hope to choose the classifier with the best tradeoff among them. Table 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.

	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 1: Effectiveness and Efficiency of Weak-Classifiers.

SVM and RF are the most effective classifiers but have the highest cost (which is consistent with previous works in the literature [10]). 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.

3.2 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 (verified in Section 3.2).

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.

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

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 of the best classifier (Transformer) per dataset (Table 3) and the effectiveness of KNN. Details of the experimental setup are given in Section 4. In Figure 2, it is possible to visually grasp a very high correlation between KNN and the best Transformer models The Person's correlation between the KNN and the best models is $r = 0.84$.

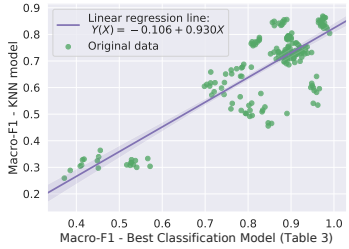

Figure 2: Correlation between KNN and Transformers models.

3.3 Time Complexity

E2SC-IS complexity is related to the KNN ($O(N^2)$, 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-IS complexity is $O(N^2)$).

4 EXPERIMENTS

4.1 Datasets

To evaluate the IS methods, we consider **19** real-world datasets (Table 2) collected from various sources in two broad ATC tasks [5, 27, 33, 34, 41]: i) *topic classification*; and ii) *sentiment analysis*.

Task	Dataset	Size	Dim.	# Classes	Density	Skewness
Topic	DBLP	38,128	28,131	10	141	Imbalanced
	Books	33,594	46,382	8	269	Imbalanced
	ACM	24,897	48,867	11	65	Imbalanced
	20NG	18,846	97,401	20	96	Balanced
	OHSUMED	18,302	31,951	23	154	Imbalanced
	Reuters90	13,327	27,302	90	171	Extremely Imbalanced
	WOS-11967	11,967	25,567	33	195	Balanced
	WebKB	8,199	23,047	7	209	Imbalanced
	Twitter	6,997	8,135	6	28	Imbalanced
	TREC	5,952	3,032	6	10	Imbalanced
	WOS-5736	5,736	18,031	11	201	Balanced
Sentiment	SST1	11,855	9,015	5	19	Balanced
	pang_movie	10,662	17,290	2	21	Balanced
	Movie Review	10,662	9,070	2	21	Balanced
	vader_movie	10,568	16,827	2	19	Balanced
	MPQA	10,606	2,643	2	3	Imbalanced
	Subj	10,000	10,151	2	24	Balanced
	SST2	9,613	7,866	2	19	Balanced
	yelp_reviews	5,000	23,631	2	132	Balanced

Table 2: Datasets Statistics

4.2 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.

[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.

Several alternative representations could be used in our experiments, including (i) **static embeddings** such as FastText [23] and (ii) **contextual embeddings** built by Transformer architectures, through a fine-tuned model or a zero-shot approach. Despite the lower dimensionality compared to TFIDF, all these representations have high density. As demonstrated in [10], the high density associated with the **static embeddings** leads to a slowing down of the classification methods on 31 times compared to TFIDF due to the high cost of processing all non-zero dimensions. Besides, [12] demonstrated that directly using **contextual embeddings** as input for IS methods is either **inefficient** (due to distance calculations with dense vectors) or **ineffective**, or both. Considering all these reasons, the **TF-IDF** representation is used as input to all IS methods, leaving for feature work the design of new IS methods that can better operate with high-density static or contextual embeddings.

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.

4.3 Text Classification Methods

As mentioned, our goal is to study and compare our proposed method against the SOTA IS techniques in the context of the **Transformer** classification approaches – *notably the SOTA in classification in several domains* [12]: **BERT** [15], **RoBERTa** [28], **DistilBERT** [40], **BART** [25], **AlBERT** [24], and **XLNet** [46].

Given the 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 methodology from [11]. 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.

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 3: Best ATC Approach by Dataset

We aim to apply the IS methods in the best possible scenario (top-best-ATC-method) for each of the considered datasets. We define as the **best approach** (by dataset), the one with the **highest** effectiveness (MacroF1) among all. We comprehensively compared all the previous Transformers approaches. Due to space limitations, we provide an online table [7] containing the results of all methods. The summary of results of the best approaches by dataset is shown in Table 3.

4.4 Instance Selection Methods

We consider as baselines a set of **six** IS methods described in Section 2, namely: *Condensed Nearest Neighbor* (**CNN**); *Instance Based 3* (**IB3**); *Local Set-based Smoother* (**LSSm**); *Local Set Border Selector* (**LSBo**); *Enhanced Global Density-based IS* (**EGDIS**); and *Curious Instance Selection* (**CIS**). All parameters for the IS methods were defined with grid-search, using cross-validation in the training set. Table 4 shows the range of parameter values for each IS method we evaluate. The best parameter in each range is marked in **bold**.

method	parameters	method	parameters
CNN	n_neighbors: [1, **3**, 5, 10]	EGDIS	n_neighbors: [1, **3**, 5, 10]
LSSm		CIS	iterations: 100*|k_cluster|
LSBo			learner: Decision Tree
IB3	Confidence Acceptance: 0.9 Confidence Dropping: 0.7		initial error: 0.5
			discount factor: 0.01
E2SC	p: 5 β (0) = δ = 0.05		epsilon: 0.9 to 0.1 (step decay)
			lr: 0.09 to 0.01 (step decay)

Table 4: Parameters of the IS methods.

4.5 Metrics and Experimental Protocol

We evaluated the instance selection methods concerning the capacity to reduce the training set, classification effectiveness and training time. 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.

According with [26], reduction mean is defined $\overline{R} = \frac{\sum_{i=0}^{k} \frac{|T_i|-|S_i|}{|T_i|}}{k}$, where T is the original training set, S is the solution set containing the selected instances by the IS method being evaluated, and k is the number of folds adopted in our experiments (10 folds).

We evaluated the classification effectiveness using Macro Averaged F1 (MacroF1)[43] 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. This method is preferred over signed-rank tests for testing hypotheses about mean effectiveness and is robust to potential violations of the normality assumption in this context[22, 44]. Finally, we applied the Bonferroni correction [21] to account for multiple tests.

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.

4.6 Experimental Results - Analyses

4.6.1 ***RQ1****. Is* ***E2SC-IS*** *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 [12]. In Table 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.

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-IS** (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.

[7] https://shorturl.at/hFJUZ

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 5: Percentage of reduction of the training set size.

The application of the IS methods to the best classifiers in each dataset (Table 3)) is seen in Table 6. The NoSel column corresponds to the results with no training set reduction. We observe in Table 6 that **E2SC-IS** 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 6: Macro-F1 - IS approaches (columns) in each dataset (rows) considering the best classifier (Table 3). Cells in bold and green background are statistically equivalent to no instance selection (NoSel).

In sum, both experiments indicate an affirmative answer for RQ1 – **E2SC-IS** 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.

4.6.2 ***RQ2****. What is the impact of applying* ***E2SC-IS*** *in the text classification models' total construction time?* Selecting only the most representative instances should, intuitively, reduce model construction time. By answering RQ1, we demonstrated that **E2SC-IS** 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 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 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. 4.5) 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 6, **E2SC-IS** achieved excellent effectiveness results and produced attractive training set reductions (on average 26.9%). As we can visually grasp, **E2SC-IS** 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-IS** achieved the best tradeoff among all methods, considering all the tripod requirements.

4.6.3 ***RQ3****. How flexible is the* ***E2SC-IS*** *framework to adjust to different application/task requirements?* As mentioned, **traditional IS strategies do not scale for the big data scenario** [12], i.e., datasets with more than 100K instances [47]. In this section, we investigate whether our solution can overcome this barrier and, if

not, whether **E2SC-IS** is flexible enough to be adapted to deal with the challenges posed by the task. In other words, we want to demonstrate that our proposal´s steps can be modified to accommodate different requirements posed by distinct (big data) scenarios.

Scenarios and Datasets. In addition to the datasets present in Table 2, 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 8 shows their statistics.

Dataset	Size	Dim.	# Classes	Density	Skewness
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 8: Large Datasets Statistics

E2SC-IS 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 other 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-IS**´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 [14], 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 reduced 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**[32], a logarithmic complexity solution. 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 9: Comparison Exact vs. Approximate KNN

Table 9 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.

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

Experimental Results. As in the previous experiments, the E2SC-IS was applied to the best classification approach in each dataset (see Table 3)). In Table 10, 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 10, 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 [12].

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

	AGNews									yelp_2013						MEDLINE					
	NoSel	E2SC#2								NoSel	E2SC#2					NoSel	E2SC#2				
		β=20%	β=25%	β=35%	β=50%	β=65%	β=75%	β=80%	β=85%		β=20%	β=25%	β=30%	β=35%	β=40%		β=20%	β=25%	β=35%	β=50%	β=65%
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)	**64.4(0.6)**	**64.2(0.4)**	**63.8(0.4)**	**63.3(0.1)**	**63.0(0.5)**	**62.4(0.6)**	**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	-	1.627x	1.708x	2.047x	2.502x	3.610x	4.761x	6.011x	7.476x	-	1.285x	1.301x	1.445x	1.551x	1.595x	-	1.452x	1.548x	1.781x	2.033x	3.304x

Table 10: Reduction-Effectiveness-Speedup Results for E2SC in Large Datasets Scenarios

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 possible 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.

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 11, we present the results regarding our two proposed instantiations of the **E2SC-IS** framework, concerning: (i) the average reduction rate; (ii) Transformer effectiveness (Macro-F1); and (iii) SpeedUps.

		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%				1.25	1.37

Table 11: Tripod Results in Small-to-Medium datasets

As Table 11 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 (RQ1). 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 (RQ2). 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 **RQ3** – **E2SC-IS** is *flexible* to adjust to different application requirements, being able to, in **all cases**, reduce the training set and maintain effectiveness, while providing significant efficiency improvements.

5 CONCLUSION

In this paper, we proposed **E2SC-IS** – a novel two-step framework aimed at large datasets with a special focus on transformers architectures. E2SC-IS 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-IS 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-IS 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 less data without losing effectiveness) and practical, allowing for savings in energy and budgets.

In the future, we will investigate how to refine our proposed heuristics for learning near-optimal reduction rates. We will also investigate how to use v-Usable Information [16] as a metric to help improve removal probabilities. We intend to introduce E2SC into AutoML solutions as a step in a data pipeline. Last but not least, it would be interesting to investigate the use of our framework in an unlabeled deep-learning pre-training stage, e.g., for building a large language model from scratch more efficiently.

ACKNOWLEDGMENTS

This work was supported by CNPq, CAPES, FAPEMIG, Amazon Web Services, NVIDIA, CIIA-Saúde, and FAPESP.

REFERENCES

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

[2] 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. 2020. "Fixing the curse of the bad product descriptions"–Search-boosted tag recommendation for E-commerce products. *Information Processing & Management* 57, 5 (2020), 102289.

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

[4] 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. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*.

[5] Jose Camacho-Collados and Mohammad Taher Pilehvar. 2018. On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis. In *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 40–46.

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

[7] Joel Luís Carbonera and Mara Abel. 2018. Efficient Instance Selection Based on Spatial Abstraction. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. 286–292. https://doi.org/10.1109/ICTAI.2018.00053

[8] Thiago Cardoso, Rodrigo Silva, Sérgio Canuto, Mirella Moro, and Marcos Gonçalves. 2017. Ranked batch-mode active learning. *Info. Sciences* (2017).

[9] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to Debias for Recommendation. In *Proc. of the ACM SIGIR Conference on Information Retrieval (SIGIR '21)*. 21–30.

[10] Washington Cunha, Sérgio Canuto, Felipe Viegas, Thiago Salles, Christian Gomes, Vitor Mangaravite, Elaine Resende, Thierson Rosa, Marcos André Gonçalves, and Leonardo Rocha. 2020. 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 (2020), 102263.

[11] 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. 2021. On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. *Information Processing & Management* 58, 3 (2021), 102481.

[12] Washington Cunha, Felipe Viegas, Celso França, Thierson Rosa, Leonardo Rocha, and Marcos André Gonçalves. 2023. A Comparative Survey of Instance Selection Methods Applied to NonNeural and Transformer-Based Text Classification. *ACM Comput. Surv.* (jan 2023). https://doi.org/10.1145/3582000

[13] Claudio M.V. de Andrade, Fabiano M. Belém, Washington Cunha, Celso França, Felipe Viegas, Leonardo Rocha, and Marcos André Gonçalves. 2023. 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 (2023), 103336. https://doi.org/10.1016/j.ipm.2023.103336

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

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*.

[16] Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding Dataset Difficulty with $\mathcal{V}$-Usable Information. In *Proceedings of the 39th International Conference on Machine Learning*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.), Vol. 162. PMLR.

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

[18] Siddhant Garg, Goutham Ramakrishnan, and Varun Thumbe. 2021. Towards Robustness to Label Noise in Text Classification via Noise Modeling. In *Proceedings of the 30th ACM International CIKM'21*. 3024–3028.

[19] Xiao Han, Yuqi Liu, and Jimmy Lin. 2021. 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*.

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

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

[22] David Hull. 1993. 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)*. 329–338.

[23] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the Conference European Chapter Association Computational Linguistics (EACL)*. 427–431.

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

[25] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th ACL*. 7871–7880.

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

[27] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S Yu, and Lifang He. 2022. A Survey on Text Classification: From Traditional to Deep Learning. *ACM Transactions on Intelligent Systems and Technology* (2022).

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

[29] Zhiwei Liu, Yingtong Dou, Philip S. Yu, Yutong Deng, and Hao Peng. 2020. 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* (Virtual Event, China) *(SIGIR '20)*. 1569–1572. https://doi.org/10.1145/3397271.3401253

[30] Washington Luiz, Felipe Viegas, Rafael Alencar, Fernando Mourão, Thiago Salles, Dárlinton Carvalho, Marcos Andre Gonçalves, and Leonardo Rocha. 2018. A feature-oriented sentiment rating for mobile app reviews. In *Proceedings of the 2018 World Wide Web Conference*. 1909–1918.

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

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

[33] Luiz Felipe Mendes, Marcos André Gonçalves, Washington Cunha, Leonardo C. da Rocha, Thierson Couto Rosa, and Wellington Martins. 2020. "Keep it Simple, Lazy" MetaLazy: A New MetaStrategy for Lazy Text Classification. In *CIKM '20*.

[34] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep Learning–Based Text Classification: A Comprehensive Review. *ACM Comput. Surv.* 54, 3, Article 62 (apr 2021), 40 pages.

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

[36] Fernando Mourão, Leonardo Rocha, Renata Braga Araújo, Thierson Couto, Marcos André Gonçalves, and Wagner Meira Jr. 2008. Understanding temporal aspects in document classification. In *Proceedings of the International Conference on Web Search and Web Data Mining, WSDM*. ACM, 159–170.

[37] Andrew Ng. 2016. Nuts and bolts of building AI applications using Deep Learning. *NIPS Keynote Talk* (2016).

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

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

[40] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

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

[42] Vishwanath A. Sindagi, Rajeev Yasarla, Deepak Sam Babu, R. Venkatesh Babu, and Vishal M. Patel. 2020. Learning to Count in the Crowd from Limited Labeled Data. In *Computer Vision – ECCV*. Cham, 212–229.

[43] Marina Sokolova and Guy Lapalme. 2009. A Systematic Analysis of Performance Measures for Classification Tasks. *Information Processing & Management (IP&M)* 45, 4 (July 2009), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

[44] Julián Urbano, Harlley Lima, and Alan Hanjalic. 2019. 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)*. 505–514.

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

[46] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NIPS*, Vol. 32. 5754–5764.

[47] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level Convolutional Networks for Text Classification. In *NIPS '16*. Vol. 28. 649–657.