Combining Active Learning with Self-train algorithm for classification of multimodal problems

Stamatis Karlos
University of Patras
Department of Mathematics
Patras, Greece
stkarlos@upatras.gr

Vasileios G. Kanas
University of Patras
Department of Electrical and Computer Engineering
Patras, Greece
vaskanas@upatras.gr

Christos Aridas
University of Patras
Department of Mathematics
Patras, Greece
char@upatras.gr

Nikos Fazakis
University of Patras
Department of Electrical and Computer Engineering
Patras, Greece
fazakis@upatras.gr

Sotiris Kotsiantis
University of Patras
Department of Mathematics
Patras, Greece
sotos@math.upatras.gr

***Abstract*—In real-world cases, handling of both labeled and unlabeled data has raised the interest of several data scientists and Machine Learning engineers, leading to several demonstrations that apply data augmenting approaches to achieve an effective learning behavior. Although the majority of them propose either the exploitation of Semi-supervised or Active Learning approaches, individually, their combination has not been widely used. The ambition of this strategy is the efficient utilization of the available human knowledge relying along with the decisions driven by automated methods under a common framework. Thus, we conduct an empirical evaluation of such a combinatory approach over three problems, related to multimodal data operating under the pool-based scenario: Gender Identification, Recognition of Offensive Language and Emotion Detection. Into the proposed learning framework, which exploits initially labeled instances with small cardinality, our results prove the benefits of adopting such kind of semi-automated approaches regarding both the achieved predictive correctness and the reduced consumption of time and cost resources, as well as the smoothness of the learning convergence, mainly using ensemble classifiers.**

Keywords—Active self-training framework, Semi-supervised learning, Extremely Randomized Trees, data augmentation techniques, semi-automated approaches

I. Introduction

Although the purpose of Machine Learning (ML) algorithms is to inject intelligent methods that mimic human behavior inside related learning frameworks, their automated operation may suffer from a series of phenomena that occur in large-scale ecosystems [1], such as the unstable character of underlying conditions or the evolvement of time-based facts – known as concept drift [2] – as well as the inability to tackle with Big Data problems under tight time constraints [3]. This kind of implications have induced a new situation in the field of ML research: instead of trying to collect vast amounts of instances, whose assignment of their target variable is usually difficult to be mined through an automated process, demanding at the same time much human effort, adoption of techniques that are based on small portions of labeled data exploiting collected or provided unlabeled data so as to refine more accurate predictive models has been widely applied the last years in real-life scenarios.

As a consequence, a large family of approaches has been devised, whose main learning core relies on augmenting the cardinality of the existing instances iteratively with the most appropriate non-annotated instances. The term "appropriate" is usually measured through a suitable ranking metric. An in-depth review of such works has been published by Schwenker and Trentin [4] categorizing this kind of approaches as Partially Supervised Learning (PSL) techniques. For the rest of this work, the case of classification task under the pool-based scenario will be theorized as the main concept of the described mechanisms. Into this context, our proposed framework deals with the trade-off of achieving accurate classification performance without spending much human effort. Thus, the term learner coincides with the meaning of classifier and the target variable is either in categorical format or in a discretized numerical one.

Active Learning (AL) category of algorithms consists of approaches that provide a semi-automated solution, since the predictive power of both the human factor and the products of ML field is combined under suitable frameworks. To be more specific, starting with a small labeled subset (L) of the total collected data (D), a selected base-learner is trained on *L* and is then applied on the rest of the data – called as unlabeled data (U) – so as to rank them according to an informativeness criterion. The above process varies based on the structure of the base-learner – for example, geometry plays crucial role in boosting the performance of Support Vector Machines (SVM) learner in [5], while the Expected Loss Optimization (ELO) [6] is more generalizable. After the detection of the most highly-ranked instances, human factor is responsible for annotating them based on its knowledge or its expertise. This role could be addressed either by human experts, regarding mainly scientific fields that demand specialized theoretical or technical background, or even by larger amount of human entities, a case that is noted as crowdsourcing. This latter case is usually met in recommendation engines, where the opinion of each individual is requested and is evaluated under favoring metrics (e.g. popularity, co-coverage) [7]. Then, the decisions exported the by human factor are arguably accepted as correct and the currently available *L* subset is enriched with the newly labeled data.

978-1-7281-4959-2/19/$31.00 ©2019 IEEE

On the contrast, Semi-supervised Learning (SSL) category does not exploit at all the human factor but is solely based on the decisions produced by the corresponding selected base-learner. Hence, instead of measuring another one quantity that transforms the output of the base-learner into a convenient form, as in case of AL, its predictions are manipulated as an adequate indication for mining the unlabeled data that would augment the corresponding *L* subset. Despite the fact that this variant of PSL approaches leads to self-confident algorithms, their learning behaviors have been proved really successful in practice [8], [9]. The two basic strategies usually met in order to reduce this inherent property of SSL algorithms are either to introduce specific thresholds or optimization procedures during the mining of confident unlabeled data [10] or to employ robust base-learners into the main learning kernel [11].

Consequently, judging by the manner that these two separate PSL categories operate, a hybrid framework could be devised in order to compromise both of them efficiently. Of course, the ambition here should be the relaxation of the human factor dependency, since this involvement more usually than not induces additional expenses and time delays, without sacrificing at the same time much of the predictive ability of the finally constructed learners [12]. Prioritizing according to this trade-off, a framework that combines AL process along with the self-training algorithm – a well-known variant of SSL algorithms – is described in this work, letting them to act interchangeably during the iterative proposed process. Moreover, its efficacy is tested specifically on raw-data with multiple modalities that concern data from sound, image and video signals as well as text-based sources [13], [14]. Since this kind of data are easily interpretable and conceivable by human factor, no restrictions are posed regarding the comprehension by the human entities.

The rest of this work is organized as follows: in Section II, related works are reported briefly, including both pioneering approaches of combining AL and SSL approaches, and some of the most recently demonstrated. Section III contains the description of the proposed framework, while the next Section gives some information about the examined multimodal datasets. Finally, our results along with some statistical comparisons and comprehensive comments are given in Section V, before we sum up in the last Section, where potential improvements are mentioned.

II. Related Work

Both AL and SSL are iterative procedures aiming to reduce the burden of manual labeling, either by finding the most informative sample in each iteration for human labeling (AL), or by exploiting the machine itself to label samples. McCallum and Nigam [15] were the first who noticed the complementarities between AL and SSL. In their work, they combined committee-based AL with EM-based SSL for text classification. Later, co-testing was proposed in [16], a variant of query-by-committee method. In this method, two different views of features were used to train two classifiers separately. Then the unlabeled instances in which the classifier disagree the most were selected for human annotation. Finally, co-testing and co-training were combined using an expectation maximization (co-EM) algorithm to automatically label instances that showed a low disagreement between the two classifiers.

In another study [17], authors proposed a unified framework using the global entropy reduction maximization criterion for speech recognition. The authors in [18] studied cross-lingual sentiment classification. They proposed a new model based on the initial training data from the source language and the translated unlabeled data from the target language. The initial training data are used to train a base classifier, which consequently is applied to the translated unlabeled data. Then AL selects the most representative examples to be labeled by a human expert. During this labelling process, the human expert evaluates the overall sentiment polarity. Simultaneously, self-training selects some of the most confident classified examples with the corresponding predicted labels, which are added to the training set for the next learning cycle. In the next cycle, the model is retrained based on the augmented training data and this process is repeated until a termination condition is satisfied.

More recently, sound classification was studied [19]. In this work, the proposed method, applied on pool-based and stream-based processing scenarios, pre-processes the unlabeled instances by calculating their confidence scores based on a classifier performance, and then the candidates with lower scores are delivered to human annotators, while those with high scores are automatically labeled by the machine. A large database of environmental sounds was collected there (about 15 hours of raw-data) where numerous features were created so as to capture numerous views of the same sound instance. Social networks have also been a field of interest for this kind of approaches, since self-training alongside active learning has been utilized for named entity recognition on Twitter [20]. More specifically, uncertainty-based and diversity-based sampling methods, used as AL query strategies, were applied to the unlabeled data to select the most informative instances, which consequently were labeled by an expert. In addition, the non-informative instances were fed to a conditional random field model and the high confident classified instances were selected. After this process both the manually-labeled and machine-labelled instances were added to the training data to retrain the classification model.

Furthermore, [21] semi-supervised active learning has been used for support vector machines, aiming to exploit the underlying structure information given by the spatial pattern of the (un)labeled data in the feature space. Probabilistic models were used to capture the data structure. These models were iteratively improved at run-time with newly available labeled data during the AL process. The probabilistic models were considered in a selection strategy based on distance, density, diversity, and distribution information for AL (4DS strategy) and in a particular kernel function for SVM (Responsibility Weighted Mahalanobis kernel).

III. Proposed Framework

The key factor of the proposed combinatory scheme is the proper exploitation of two different PSL approaches obtaining the benefits from both sides and reconciling successfully the emerging trade-off between achieved accuracy and employment of human effort. Hence, our ambition is to incorporate into the learning kernel of the proposed iterative process the human factor much less than the selected algorithm of SSL, deploying a combined approach that competes ideally the individual AL approach. Thus, the costly and usually time delaying manual annotation of human factor should be reduced. Hereinafter, this quantity will be mentioned as *al_ssl_ratio,* representing the fraction of the instances that should be annotated by the two different mechanisms into the combined approach. The notation of H_f would also be used when we refer to the human factor.

Notwithstanding the small participation of H_f, its knowledge could both boost the total performance of the proposed framework, by applying discriminative query strategies (Q_{st}) that extract meaningful unlabeled instances (u_i) from the corresponding *U* pool, so as to be provided for the labeling stage – instead of using just the confidence of base-learner [19] – and at the same time control the amount of instances that would be totally mined, keeping it small enough for producing feasible solutions in real-life scenarios.

As it concerns the part of SSL approach, self-train algorithm was preferred to be integrated into the proposed framework, as one of the most representative and well-studied product of this category [22]. This wrapper algorithm is based solely on the model that is initially built on the provided labeled pool of data with the prerequisite that the selected base-learner belongs to the family of probabilistic learning models. Based upon this assumption, for each u_i a vector of class probabilities is exported whose dimension is *cl x 1*, where the *cl* parameter depicts the predefined number of classes that appear into each examined dataset. Then, the class with the largest class probability is assigned to the i-th unlabeled instance and is transferred into the *L* subset, whose cardinality is now growing. Although various criteria have been implemented for avoiding mislabeling errors during the phase of accepting or rejecting the decisions of the base learner, such as threshold values, similarity measures or distance metrics [23], it has been preferred here not to insert anyone of these mechanisms. In this way, not additional overhead time expenses are introduced, letting the self-train algorithm to operate under a simplistic and self-confident version.

Furthermore, without re-training or applying any exhaustive searches, we neither increase computational complexity of total framework nor reach to the point of using heuristics methods for compromising all posed restrictions. This fact favors the smooth consumption of the total budget (B) that is inserted as one of the main parameters into our learning framework and enables the proper comparison of any produced variants. The corresponding pseudo-code along with the needed input variables is given in Figure 1. In Figure 2 is also placed the pseudo-code of a necessary function during the preprocessing stage of the proposed framework. Moreover, for discriminating the produced variants of this framework, a favorable notation that encompasses all the necessary input quantities could be used as follows: *AL_SelfTrain (base*$_{lea}$*, Q*$_{stg}$*, B, al_ssl_ratio, steps)*. Regarding also the convenience that our framework offers, only small modifications are needed so as to obtain just an AL and the corresponding SelfTrain approach that consume the provided budget parameter with the same way that the proposed combined version does. Of course, in the latter case the argument of Q_{stg} is unnecessary.

Therefore, these two families of counterparts could denote the quality of the *AL_SelfTrain* framework for any given base$_{lea}$ holding the rest of the parameters same. To be more specific, the combined framework should outreach the similar approaches based on SSL concept, since no human intervention takes place, rendering it as the most inexpensive solution. On the contrary, the ambition is to ensure as much closer performance – measured by appropriate metrics – of the *AL_SelfTrain* with the approaches that stem solely on AL

```
Framework AL_SelfTrain
Mode:
Pool-based scenario over a provided dataset ($D_{(f+1)\,x\,n}$)
$\{x_i , y_i\}$ – i-th instance of $D_{(f+1)\,x\,n}$ with $1 \le i \le n$
x – vector with f features
y – scalar variable depicting the categorical class
Input:
$L^0$ – initially collected labeled instances, $L^0 \subset D$
$U^0$ – initially collected unlabeled instances, $U^0 \subset D$
$L^k$ – labeled instances during k-th iteration, $L^k \subset D$
$U^k$ – unlabeled instances during k-th iteration, $U^k \subset D$
base$_{lea}$ – selected base classifier
$Q_{stg}$ – applied query strategy based on base$_{lea}$
B – Number of unlabeled instances to get labeled
al_ssl_ratio – fraction of AL and SSL participation in labeling process
steps – size of batches from instances to be labeled per iteration
$H_f$ – employed human factor or crowdsourcing platform
iters – number of combined executed iterations
Preprocess:
ALinst, SSLinst, iters = Compute_instances_per_iter (B, al_ssl_ratio, steps)
Main Procedure:
Set k = 0
While iters > 0 do
    # Active Learning part
    Train/Update base$_{lea}$ on $L^k$
    Rank through $Q_{stg}$ all $u_i \in U^k$
    Detect from $U^k$ the indices (ALind) of the top-ALinst instances
    Provide them to $H_f$ and assign its decisions to their class value
    $B := B – ALinst$
    Update $L^k$: $L^{k+1} \leftarrow L^k \cup \{x_j , H_f(x_j)\}$ for each $j \in$ ALind
    Update $U^k$: $U^{k+1} \leftarrow U^k \setminus \{x_j\}$ for each $j \in$ top-ALind
    $k := k +1$
    # Self-train part
    Train/Update base$_{lea}$ on $L^k$
    Compute class probabilities through base$_{lea}$ for all $u_i \in U^k$
    Detect from $U^k$ the indices (SSLind) of the top-SSLinst most certain
    instances and assign the corresponding class value
    $B := B – SSLinst$
    Update $L^k$: $L^{k+1} \leftarrow L^k \cup \{x_j , \text{argmax}_{Cl} P(y_j | x_j)\}$ for each $j \in$ SSLind
    Update $U^k$: $U^{k+1} \leftarrow U^k \setminus \{x_j\}$ for each $j \in$ SSLind
    $iters := iters + 1$
Output:
Use base$_{lea}$ trained on $L^{iters}$ to predict class labels of test data.
```

Fig. 1. The combined framework of AL_SelfTrain.

```
Function Compute_instances_per_iter (B, al_ssl_ratio, steps)
Restrictions:
B, steps and iters arguments should be integers
al_ssl_ratio should be expressed as a fraction of integers: Nom/Denom
Main Procedure:
Obtain Nom and Denom
ALinst = Nom * steps
SSLinst = Denom * steps
iters = B / ((Nom + Denom) * steps)
Output:
Return ALinst, SSLinst and iters quantities
```

Fig. 2. The pseudo code of the Compute_instances_per_iter function.

concept, since the scenario of surpassing this counterpart would be an ideal case. Thus, achieving similar learning behaviors with small *al_ssl_ratio* values is the most important factor, because only a quota of the demanded human effort by the latter approach is asked during the operation of the former one. Moreover, Random Sampling (RndS) process should be also inserted as an alternative query strategy, settling the baseline rival from the view of AL algorithms.

IV. Examined Datasets

In this Section, a brief description of the most important properties per each examined multimodal dataset is given. We considered three of them out of the corresponding data repositories, so as to capture the most prominent sources of raw-data: speech signal, video signal – which consists of consecutive image frames accompanied by sound signal – and text sources. Moreover, among a large variety of related datasets, crucial role played both the publication date of them – trying to choose recently demonstrated works – and the fact of being publicly available.

A. *Gender Identification (Voice)*

The current dataset refers to the gender's identification of examined speakers using speech samples. Although this problem is easily solved through physical means, its fulfillment with ML approaches demands appropriate digital signal and feature engineering processing so as to reveal patterns that could discriminate the male and female categories. In our case, 3.168 speech samples, were produced and pre-processed by a suitable package [24] that enables the measurement of acoustic quantities (e.g. mean frequency, standard deviation of frequency, spectral entropy and flatness). Duration of each sample has been fixed equal to 20 seconds, while peak frequency was omitted from the final constructed dataset. Hence, 20 features remain for fitting any predictive model for the included instances [25].

Moreover, the cardinality of each class is the same, leading to a perfectly balanced binary-class problem. Regarding the difficulty of this task, a simple acoustic model approach of the underlying properties that hold, may lead to really poor performance without tuning frequency thresholds, a process that may be difficult for the following two scenarios: i) when much more examples are given tuning would be computationally expensive ii) when just a small portion of data is provided, since the variance of the examined variable might not capture efficiently the new instances whose behavior would be unknown.

B. *Recognition of offensive language (HateSpeech)*

The problem that was tackled by the authors of [26] is related with the context of phrases that are posted on social media, and to be more specific, in Twitter, a platform that is mainly based on statements that contain text raw-data. Thus, a classifier that recognizes initially the hate-speech phrases from the neutral ones is implemented, and at the same time, separates the former to just offensive language and language that expresses pure hate. Some examples of this last category are phrases that refer to groups of people in derogatory terms, insult them or even threaten violence. Consequently, the current structure of this dataset supports a multi-class problem with three distinct classes, where the accurate separation of the last two classes seems the most desirable expectation, since phenomena that are combined with hate speech on social platforms are judged nowadays more and more as illegal and should be prevented or detected efficiently by the underlying mechanisms for reassuring a high-level quality of user interaction with each platform or even assign the appropriate penalties to culpable persons.

The main points of the creation of this dataset was the extraction of tweets made by 33.458 users that contain words related with hate-speech based on an appropriate lexicon. Since the volume of this set was about 100 million separate instances, a random sampling process was applied so as to remain only 25.000 of them. Then, a crowdsourcing process took place, where at least three human-entities were asked to categorize each one of these tweets into the most prominent of the three pre-defined classes. Hopefully, the agreement rate between the crowdsourcing entities was high enough. After this annotation stage, only the tweets whose no class did not collect the majority of the human votes were discarded, leading finally to a dataset with 24.783 instances/tweets, where the quota of each class is neutral-77.43%, offensive language-16.8% and hate speech-5.77%. As it concerns the created features, some common text-based quantities were mixed with some variables that are oriented towards both the sentimental view of phrases of social media domain and the specific properties of Twitter platform. In total, 170 features were kept for constructing the final dataset.

More informative comments about the reasons that these 3 classes confuse both human annotators and automated classifiers are demonstrated in the original publication, such as the absence of some keywords or expressions that trigger the existence of insulting or racist context.

C. *Detection of emotion (ANAD)*

This kind of dataset is related with the emotion expressed through speech signals that are extracted from videos of Arabic talk shows. Although similar works have been accomplished for various languages, only recently this dataset came up concerning Arabic corpus [27]. Instead of using just a text-based solution that does not reveal any clue about the emotional situation of any speaking entity, causing possible misunderstandings when the meaning of a sentence is implicit. Apart from applications where deaf people could be favored to communicate accurately with their co-speakers, emerging tasks such as the adoption of anchors in media, could be enhanced regarding their quality of service.

As it concerns the creation and annotation of this dataset, a small amount of video signals was initially recorded and afterwards, were provided to 18 listeners. Their task was to decide about the prevailing emotional situation of participants among these of angry, happy and surprised. After removing some specific segments from the raw-data, all the rest were divided into chunks with duration equal to 1 second. Eventually, 1384 instances were created, where the quota of each class is 53.58%, 36.5% and 9.9%, respectively. The features that were used are mainly based on 25 low-level acoustical features (e.g. MFCC, ZCR) and a number of variables that are produced applying some well-known statistical functions over these. The final amount of the remaining features sums up to 844.

V. Experiments and Results

This Section describes the experimental procedure that was executed so as to implement proper comparisons among the algorithms produced by the proposed framework, its two main variants, the baseline method of AL concept, as well as one similar approach embedded into the aforementioned framework. Before reporting these, we have to define the selected query strategies. Actually, Uncertainty Sampling Q_{stg} (UncS) has been preferred in the context of this work, as one of the most widely used and easily applicable in the literature [28]. This choice enables the creation of several versions of the same strategy, trying to find the most uncertain instance according to the predictions of the $base_{lea}$ and the selected measure of uncertainty. The three preferred versions of UncS strategy are the following:

- Entropy (Ent), a popular formula, which measures the average information revealed by any examined variable. Its general form sums up the *– zlog(z)* quantity for each class and selects this that induce the maximum information, where z is replaced by the *a posteriori* probability: *P(y|x)*,

- Smallest Margin (Mrg), a metric that translates the sense of uncertainty into the closeness of the two largest likelihoods between the contained classes. Thus, the smaller is this value, the most ambiguous is the behavior of the $base_{lea}$ according to this instance and has then to be extracted so as to be annotated by H_f,

- Minimum Standard Deviation (Std), the well-known mathematical function that takes into consideration the *a posteriori* probabilities for all classes per instance and the smaller this value is, the more uncertain is the $base_{lea}$ about this instance.

Hence, the 9 separate PSL approaches that would be composed here, independently of the parameters apart from Q_{stg}, could be summarized as follows: i) three combined approaches: AL_SelfTrain(Ent), AL_SelfTrain(Mrg) and AL_SelfTrain(Std), ii) three pure AL approaches: AL(Ent), AL(Mrg) and AL(Std), iii) the default Self-training (SelfTrain), iv) the baseline of AL concept AL(RndS) that provided randomly selected instances to H_f and v) a hybrid of the proposed framework, where the AL(RndS) and SelfTrain act interchangeably under the same scheme. In order to provide more comprehensible notations of the already mentioned algorithms, we just recorded the metric under the UncS strategy, while in case of Random Sampling we used a suitable abbreviation of the query strategy.

Regarding the rest of the involved parameters, and taking into consideration the restriction that is posed by the function of Fig. 2 about the integer format of the input arguments, the next set of values has been selected: *steps* ∊ {2, 5, 10, 20}, while the pair of (*B, al_ssl_ratio*) ∊ {(160, 1/3), (200, 1/4)}. Hence, the number of combined iterations for both cases of (*B, al_ssl_ratio*) pair is 20, 8, 4, and 2, analog to the value of *steps* parameter, where each combined iteration consists of one iteration of AL and SSL mechanisms. It is evident that per each actively labeled batch of instances by H_f, the batches that are assessed by the upcoming SSL algorithm are three or four times larger, reducing the spent human effort compared with the simple AL approach by 75% and 80%, respectively.

Although the selected values of Budget parameter seem quite small, they indeed keep pace with the similarly small initial cardinality of labeled subsets (L^0). More specifically, the three examined datasets were split to train and test subsets, covering the 90% and 10% of the total dataset, respectively. Then, the train dataset (D ≡ L ⋃ U) is divided into L and U subsets according to Labeled Ratio parameter – here mentioned as *R* and measured in percentage values – whose value is usually small enough for simulating the scarce of labeled data. Its formula is shown in next equation:

$$R\ (\%) = cardinality(L)\ /\ cardinality(D) \qquad (1)$$

The values of *R* during our experimental procedure were equal to 1% and 5%. The cardinalities of the corresponding L, U and test subsets for all our evaluated datasets are presented here:

TABLE I. REPRESANTITIVE QUANTITIES OF EXAMINED DATASETS

Datasets	Properties			
	Features	*Train instances*		*Test instances*
		R = 1%	R = 5%	

Voice	20	L = 28
		U = 2823
			L = 142
			U = 2709
				317

HateSpeech	170	L = 223
		U = 22.081
			L = 1115
			U = 21.189
				2479

ANAD	844	L = 12
		U = 1232
			L = 62
			U = 1182
				139

Summing up all the constructed scenarios, there exist three datasets, examined under two different *R* values, two separate combinations of applying the synergy of AL and SSL mechanisms consuming the provided Budget. This leads to 12 (3x2x2) cases, where each one operates under four distinct step-based approaches. The last parameter that has to be selected is this of $base_{lea}$. Five different classifiers have been contained for evaluating their learning behavior under the proposed framework:

- Extremely Randomized Trees (ExT): an ensemble learner that fits several unpruned trees over various subsamples of the provided data, aggregating their decisions for achieving accurate predictions [29],

- Random Forest (Rf): an ensemble learner which is differentiated mainly by the ExT because of the resampling process during the formatting process of the decision trees, since each subsample is chosen through replacement [30],

- Multi-Layer Perceptron (MLP): a typical neural network with one layer of 100 neurons that uses stochastic gradient decent method for weight optimization [31],

- k-Nearest Neighbors (kNN): a well-known lazy classifier that applies a voting stage of the decision of the of k closer instances to any test example [32] and

- Naive Bayes (NB): the popular learner that is based on Bayes' Theorem and is exporting the class that maximizes the maximum a posteriori hypothesis [33].

Moving to more technical details, all the included learners are adopted with their default values from sklearn Python package [34]. Therefore, kNN will be symbolized as 5NN, hereinafter. Moreover, all the L^0 subsets were formatted through stratified sampling process and all the experiments were repeated three times. The main performance metrics for our experiments have been selected to be the classification accuracy (acc) and the f1-score. This last metric constitutes a weighted average of precision and recall, two more widely assessed metrics by ML community, which depict the exactness and completeness of any tested classifier. However, f1-score is a great solution for leveraging the importance of recorded results over imbalanced datasets [35]. Similar results are also produced for the rest two metrics, using our Python-based implementation that is inspired by [36].

In order to verify under which *step* value the different learning concepts are better favored, we have measured the difference of both used metrics between the final iteration and the initial stage per each classifier (improvement). Then, we computed the maximum average difference and the minimum standard deviation. The best step is recorded, along with its winning frequency inside the corresponding parentheses. The extracted results are provided in Table II. It has also to be mentioned that the H_f has been replaced during all our results from an ideal oracle that always predicts correctly the label of each requested instance.

TABLE II. IDENTIFICATION OF THE MOST FAVORABLE STEP VALUE

Dataset	**acc**		**f1-score**	
Voice	***improvement***	***stability***	***improvement***	***stability***
AL_SelfTrain	2 (40%)	2 (40%)	2 (37.5%)	2 (40%)
AL	2 (42.5%)	5 (35%)	2 (38.75%)	2 (32.5%)
SelfTrain	2 (40%)	2 (45%)	2 (40%)	2 (40%)
HateSpeech				
AL_SelfTrain	5 (35%)	5 (26.25%)	5 (38.75%)	5 (28.75%)
AL	2 (31.25%)	20 (32.5%)	2 (30%)	20 (26.25%)
SelfTrain	5 (45%)	2 (35%)	5 (45%)	10 (30%)
ANAD				
AL_SelfTrain	2 (35%)	20 (27.5%)	2 (32.5%)	20 (30%)
AL	2 (33.75%)	10 (30%)	2 (30%)	5 (30%)
SelfTrain	10 (35%)	10 (40%)	20 (40%)	10 (55%)

It is evident that the value of step equal to 2 is the most favorable, as it concerns both acc and f1-score metrics. Actually, the half of all the cases (18 out of 36) recorded the smallest of the step values as the winning one, while the second smallest – step equal to 5 – follows with 8 victories. These results are quite reasonable, especially in cases human oracle is injecting its knowledge, since larger amount of iterations are conducted and a new refined model is created each time. Thus, providing that accurate predictions are made during the labeling process and no over-fitting phenomena appear, new insights may be revealed by the exploited learner and different unlabeled instances could be targeted so as to increase the total predictive power of the whole algorithm.

TABLE III. ACHIEVED CLASSIFICATION ACCURACY FOR EXT CLASSIFIER WITH R = 5% OVER ANAD DATASET PER STEP

Algorithm	**Accuracy (%)**			
	Step2	**Step5**	**Step10**	**Step20**
AL_SelfTrain(Ent)	81.77	82.73	82.01	82.97
AL_SelfTrain(Mrg)	89.45	89.69	88.73	89.69
AL_SelfTrain(Std)	89.93	88.97	89.21	88.25
AL_SelfTrain(RndS)	84.41	83.93	84.41	83.21
AL(Ent)	83.93	82.49	82.25	82.01
AL(Mrg)	95.20	95.68	96.40	94.72
AL(Std)	93.53	92.33	93.77	92.33
AL(RndS)	88.97	87.53	87.05	88.73
SelfTrain	81.06	82.25	81.30	80.10
Initial accuracy	81.54	**Full training accuracy**		93.63

The impact also of each query strategy has to be measured. For this reason, instead of just recording the achieved improvement between the subset of labeled data formatted after the final iteration and the starting one, proper statistical comparisons that take into consideration the achieved values per each iteration and for all applied step values have been executed. The first stage includes the application of Friedman statistical test [37], in order to obtain the related ranking among all the 9 approaches per Q_{stg} and $base_{lea}$. Secondly, a post-hoc test of Nemenyi [38] is applied so as to ascertain the statistical importance of the obtained behaviors. Due to lack of space, for facilitating the presentation of these results, only a small portion of them are demonstrated here, while the rest have been placed along with our code implementation in the following link: https://github.com/vaskanas/ke80537/tree/master/iisa2019.

Table III depicts the achieved accuracy per each *step* value, compared also with the supervised scenario – where all the existing data are used for building a predictive model – only for one out of the 12 cases: the ANAD dataset, regarding ExT classifier and labeled ratio equal to 5%. The names of the algorithms have been shortened for viewability reasons, while their notation should be AL_SelfTrain(ExT, Q_{stg}, 160, 1/3, steps). The improvement that is offered by the combined approach is about 8% and 5% against its main rivals – SelfTrain and AL_SelfTrain(RndS) – while smaller improvement is recorded compared with AL(RndS), an approach that demands 4 times larger amount of labeled instances by human. Additionally, these results prove also the better quality of Mrg and Std strategies, which has been generalized also in the majority of the conducted experiments.

Of course, the fact that these results are produced with ExT built only on 62 labeled instances and demands additionally only another 40 instances constitutes its main asset, being harmonized with real-life scenarios where only small initial labeled data have been manually gathered and simultaneously, the consumption of human effort is highly restricted due to occurring expensed and delays. Moreover, its performance based on small *L* subsets allows us to make use of validation set that would enable the detection of the best achieved score during all the spectrum of conducted iterations. This optimization stage is left as a future task [39].

In the sequel, for each Q_{stg} and per distinct classifier, we count the frequency of the cases that the ranking of the examined query strategy under the proposed framework is higher than: the same AL approach with the same query strategy, the baseline of AL concept, the hybrid approach (AL_SelfTrain(RndS)) and SelfTrain. These results are placed in Table IV. Three out of the five examined learners (ExT, Rf and NB) have been highly benefited against their ri-

TABLE IV. FREQUENCY COUNT OF PROPOSED FRAMEORK VICTORIES CONCERNING STATISTICAL RANKING

Classifier	**Victories (acc/f1-score)**			
ExT	***AL(Qstg)***	***AL(RndS)***	***AL_SelfTrain (RndS)***	***SelfTrain***
Ent	4/4	0/0	2/2	8/9
Mrg	0/0	11/11	12/12	12/12
Std	0/0	6/8	11/11	10/11
Rf				
Ent	5/5	0/0	1/2	10/9
Mrg	0/0	8/11	12/12	12/12
Std	0/0	5/8	8/11	11/11
MLP				
Ent	2/0	0/0	10/7	10/8
Mrg	2/0	6/3	10/8	9/7
Std	2/0	1/0	9/7	9/7
5NN				
Ent	3/4	1/2	7/6	9/9
Mrg	1/2	2/1	5/7	8/8
Std	1/2	2/2	4/7	9/8
NB				
Ent	4/7	4/5	6/8	9/9
Mrg	2/4	8/9	12/12	12/2
Std	1/4	8/9	12/12	12/12

vals, especially when Q_{stg} is either Mrg or Std, while the rest learners have mainly outperformed SelfTrain and the hybrid approach, without performing well compared with AL(RndS). The main reason why this happen is the dependence of these two learners by their parameters, which in our work have been set to their default values. However, a tuning procedure could probably boost their performance [40]. On the other hand, the learners based on decision trees managed to record great learning behaviors under both small labeled ratio scenarios, proving that their ensemble strategy can provide reliable solutions.

Finally, the next Table contains the ranking of all the approaches related with the HateSpeech dataset, when the parameters {R, bese_{lea}, B, al_ssl_ratio} coincide with the tuple of {1%, ExT, 160, 1/3}. The algorithms that are marked with a sign do not differ statistically, according to the applied post-hoc test, with significance level equal to 0.05. From this point of view, Std-based approach that stems from the proposed framework did not perform as well as in the case of ANAD dataset, but the corresponding Mrg-based algorithm was ranked in the 2nd position, above all its main rivals, without performing statistically different than the corresponding AL approach, which uses four times greater amount of human annotated instances. The total volume of rankings is placed in the abovementioned link.

TABLE V. FRIEDMAN RANKING OF ALL 9 APPROACHES OVER HATESPEECH DATASET

Table Head	**Table Column Head**	
	acc	***f1-score***
AL(Mrg)	12.46	12.72*
AL_SelfTrain(Mrg)	13.76*	13.29*
AL(Std)	14.00*	13.36*
AL(Ent)	15.86	19.86
AL(RndSt)	19.33*	15.10
AL_SelfTrain(RndS)	19.93*	20.67
SelfTrain	20.79	23.72
AL_SelfTrain(Ent)	23.58	21.16
AL_SelfTrain(Std)	31.05	30.88

VI. CONCLUSIONS

In this paper, we proposed a framework of combining Active Learning concept along with Self-training algorithm, oriented towards reducing the human burden over real-life scenarios, where abundant unlabeled data are usually easily collected, in contrast with labeled examples whose labeling stage either demands expert's knowledge or contribution by larger groups of human, which have to be motivated by related rewards. Hence, relaxation of this necessity is the main ambition here, trying to achieve at the same time better learning behavior, compared at least with the baseline of AL and the corresponding SSL approaches. Additionally, through augmenting the cardinality of the initially collected L subset through two different aspects, the obtained learning behavior could be boosted towards accurate predictions, competing the AL scenario that demands more human annotations than the proposed.

The constructed framework constitutes a straightforward implementation of this combination – which is highly appreciated the last years as a really effective solution by the ML community – depending on a small amount of parameters, so as to tune the consumption of the provided budget properly and according to user's choices. During its operation it supports any probabilistic classifier, since the described operation of the mechanisms that mine the unlabeled pool of instances need class probabilities for formulating the appropriate decisions. Three different datasets that contain feature spaces based on several modalities were evaluated in this context, leading to useful results about the quality of learning behavior, regarding the size of the batches that are extracted per iteration, the different metrics into Uncertainty Sampling strategy and five examined learners.

Potential improvements of this work could be the employment of deep learning networks [41] into the kernel of the proposed framework, especially during the concept of SSL part, where the selection of unlabeled instances is usually relied purely on the confidence of the base learner. The factor of interpretability should then been reviewed [42], since many of these algorithms sacrifice this property against boosting their accuracy. Different SSL approaches could also be integrated into the proposed framework, such as multi-view schemes that seem to be compatible enough with the nature of multimodal data [43], or approaches that exploit mechanisms for detecting noisy unlabeled instances [44].

Moreover, different kind of Query Sampling strategies could be exploited, since UncS, although it favors the time feasibility, often is rendered as a myopic approach. Query-By-Committee solution, which applies a voting scheme over the decisions of the participating learners, seems a hopeful solution, enabling also the use of non-probabilistic learners [45]. Finally, design of query strategies that try to optimize more than on criterion so as to tackle with the efficient ranking of unlabeled examples is an active field [46].

ACKNOWLEDGMENT

This research is implemented through the Operational Program Human Resources Development, Education and Lifelong Learning and is co-financed by the European Union (European Social Fund) and Greek national funds.

REFERENCES

[1] C. Wang, A. Kalra, L. Zhou, C. Borcea, and Y. Chen,

"Probabilistic Models for Ad Viewability Prediction on the Web," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 2012–2025, Sep. 2017.

[2] I. Khamassi, M. Sayed-Mouchaweh, M. Hammami, and K. Ghédira, "Discussion and review on evolving data streams and concept drift adapting," *Evol. Syst.*, vol. 9, no. 1, pp. 1–23, Mar. 2018.

[3] S. Shayaa *et al.*, "Sentiment analysis of big data: Methods, applications, and open challenges," *IEEE Access*, vol. 6, pp. 37807–37827, 2018.

[4] F. Schwenker and E. Trentin, "Pattern classification and clustering: A review of partially supervised learning approaches," *Pattern Recognit. Lett.*, vol. 37, no. 1, pp. 4–14, 2014.

[5] W. Liu, L. Zhang, D. Tao, and J. Cheng, "Support vector machine active learning by Hessian regularization," *J. Vis. Commun. Image Represent.*, vol. 49, pp. 47–56, 2017.

[6] B. Long, J. Bian, O. Chapelle, Y. Zhang, Y. Inagaki, and Y. Chang, "Active learning for ranking through expected loss optimization," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1180–1191, 2015.

[7] M. Elahi, F. Ricci, and N. Rubens, "A survey of active learning in collaborative filtering recommender systems," *Comput. Sci. Rev.*, vol. 20, pp. 29–50, 2016.

[8] P. Kang, D. Kim, and S. Cho, "Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing," *Expert Syst. Appl.*, vol. 51, no. C, pp. 85–106, Jun. 2016.

[9] M. K. Dalal and M. a. Zaveri, "Semisupervised Learning Based Opinion Summarization and Classification for Online Product Reviews," *Appl. Comput. Intell. Soft Comput.*, vol. 2013, pp. 1–8, 2013.

[10] D. Wu, X. Luo, G. Wang, M. Shang, Y. Yuan, and H. Yan, "A Highly Accurate Framework for Self-Labeled Semisupervised Classification in Industrial Applications," *IEEE Trans. Ind. Informatics*, vol. 14, no. 3, pp. 909–920, 2018.

[11] S. Karlos, N. Fazakis, S. Kotsiantis, and K. Sgarbas, "Self-train logitboost for semi-supervised learning," in *EANN 2015. Communications in Computer and Information Science*, 2015, vol. 517.

[12] T. Sabata, P. Pulc, and M. Holena, "Semi-supervised and Active Learning in Video Scene Classification from Statistical Features," in *IAL@PKDD/ECML*, 2018, vol. 2192, pp. 24–35.

[13] R. Potapova and V. Potapov, "On Individual Polyinformativity of Speech and Voice Regarding Speakers Auditive Attribution (Forensic Phonetic Aspect)," in *Speech and Computer. SPECOM. Lecture Notes in Computer Science, vol 9811.*, 2016, pp. 507–514.

[14] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, "Multimodal Machine Learning: A Survey and Taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, 2019.

[15] K. McCallumzy Andrew Kachites;Nigamy, "Employing EM and pool-based active learning for text classification," in *ICML*, 1998, pp. 350–358.

[16] I. Muslea, S. Minton, and C. A. Knoblock, "Active+ semi-supervised learning = robust multi-view learning," in *ICML*, 2002, pp. 435–442.

[17] D. Yu, B. Varadarajan, L. Deng, and A. Acero, "Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion," *Comput. Speech Lang.*, vol. 24, no. 3, 2010.

[18] M. S. Hajmohammadi, R. Ibrahim, A. Selamat, and H. Fujita, "Combination of active learning and self-training for cross-lingual sentiment classification with density analysis of unlabelled samples," *Inf. Sci. (Ny).*, vol. 317, pp. 67–77, 2015.

[19] W. Han *et al.*, "Semi-supervised active learning for sound classification in hybrid learning environments," *PLoS One*, vol. 11, no. 9, pp. 1–23, 2016.

[20] V. C. Tran, N. T. Nguyen, H. Fujita, D. T. Hoang, and D. Hwang, "A combination of active learning and self-learning for named entity recognition on Twitter using conditional random fields," *Knowledge-Based Syst.*, vol. 132, pp. 179–187, Sep. 2017.

[21] A. Calma, T. Reitmaier, and B. Sick, "Semi-supervised active learning for support vector machines: A novel approach that exploits structure information in data," *Inf. Sci. (Ny).*, vol. 456, pp. 13–33, Aug. 2018.

[22] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowl. Inf. Syst.*, vol. 42, no. 2, pp. 245–284, Feb. 2015.

[23] M. Li and Z. Zhou, "SETRED : Self-training with Editing," *LNAI*, vol. 3518, pp. 611–621, 2005.

[24] M. Araya-Salas and G. Smith-Vidaurre, "warbleR: an r package to streamline analysis of animal acoustic signals," *Methods Ecol. Evol.*, vol. 8, no. 2, pp. 184–191, 2017.

[25] "Gender Recognition by Voice | Kaggle." [Online]. Available: https://www.kaggle.com/primaryobjects/voicegender.

[26] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," in *ICWSM*, 2017, pp. 512–515.

[27] S. Klaylat, Z. Osman, R. Zantout, and L. Hamandi, "Arabic Natural Audio Dataset, v1," *Mendeley Data,* 2018. [Online]. Available: https://data.mendeley.com/datasets/xm232yxf7t/1.

[28] B. Settles, *Active Learning*, vol. 6, no. 1. Morgan & Claypool Publishers, 2012.

[29] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

[30] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," Feb. 2015.

[32] Y. Cai, D. Ji, and D. Cai, "A KNN Research Paper Classification Method Based on Shared Nearest Neighbor," *Proc. 8th NTCIR Work. Meet. Eval. Inf. Access Technol. Inf. Retrieval, Quest. Answering Cross-Lingual Inf. Access*, pp. 336–340, 2010.

[33] H. Chen, W. Liu, and L. Wang, "Naive Bayesian Classification of Uncertain Objects Based on the Theory of Interval Probability," *Int. J. Artif. Intell. Tools*, vol. 25, no. 3, pp. 1–31, 2016.

[34] L. Buitinck *et al.*, "API design for machine learning software: experiences from the scikit-learn project," in *CoRR abs/1309.0238*, 2013.

[35] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS One*, vol. 10, no. 3, pp. 1–21, 2015.

[36] O. Cohen, "Active Learning Tutorial." 2018.

[37] I. Rodríguez-Fdez, A. Canosa, M. Mucientes, and A. Bugarín, "STAC: a web platform for the comparison of algorithms using statistical tests," in *FUZZ-IEEE*, 2015, pp. 1–8.

[38] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric statistical methods.*, 3rd ed. 2013.

[39] I. Tsamardinos, A. Rakhshani, and V. Lagani, "Performance-Estimation Properties of Cross-Validation-Based Protocols with Simultaneous Hyper-Parameter Optimization," *Int. J. Artif. Intell. Tools*, vol. 24, no. 5, 2015.

[40] S. Karlos, K. Kaleris, and N. Fazakis, "Optimized Active Learning Strategy for Audiovisual Speaker Recognition," in *SPECOM*, 2018, vol. 11096, pp. 281–290.

[41] Y. Qin, R. Langari, Z. Wang, C. Xiang, and M. Dong, "Road excitation classification for semi-active suspension system with deep neural networks," *J. Intell. Fuzzy Syst.*, vol. 33, no. 3, pp. 1907–1918, 2017.

[42] M. M. G. L. I. Del Carmen Grau Garcia D. Sengupta and A. Nowé, "Interpretable self-labeling semi-supervised classifier," in *Proceedings of the 2nd Workshop on Explainable Artificial Intelligence*, 2018.

[43] G. Kostopoulos, S. Kotsiantis, O. Ragos, and T. N. Grapsa, "Early dropout prediction in distance higher education using active learning,", *IISA*, 2017, pp. 1–6.

[44] I. Triguero, J. A. Sáez, J. Luengo, S. García, and F. Herrera, "On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification," *Neurocomputing*, vol. 132, pp. 30–41, 2014.

[45] S. Kee, E. del Castillo, and G. Runger, "Query-by-committee improvement with diversity and density in batch active learning," *Inf. Sci. (Ny).*, vol. 454–455, pp. 401–418, 2018.

[46] T. Collet and O. Pietquin, "Optimism in Active Learning," *Comput. Intell. Neurosci.*, vol. 2015, pp. 1–17, Nov. 2015.