Query Strategies, Assemble! Active Learning with Expert Advice for Low-resource Natural Language Processing

Vânia Mendonça*, Alberto Sardinha*, Luísa Coheur*, Ana Lúcia Santos†
*INESC-ID, *Instituto Superior Técnico, †CLUL, †Faculdade de Letras
Universidade de Lisboa
Lisbon, Portugal
{vania.mendonca, jose.alberto.sardinha, luisa.coheur}@tecnico.ulisboa.pt, als@letras.ulisboa.pt

***Abstract*—Active learning plays an important role in low-resource scenarios, i.e., when only a small amount of annotated instances is available. However, one does not know what is the best active learning strategy before actually testing a handful of strategies on a labeled set, which might not be viable in a real world low-resource scenario. Instead, it would be desirable to dynamically obtain the results from the best strategy on a given scenario, while using as little annotated resources as possible.**

In this paper, we present a novel application of prediction with expert advice to combine different query strategies as experts, giving a greater weight to those which select the most useful instances. We evaluated our approach in two Natural Language Processing (NLP) tasks: Part-of-Speech tagging (for English) and Named Entity Recognition (for Portuguese). Results show that our solution keeps up with the results of the best strategy in each scenario, nearly reaching fully supervised performance with only half of the annotated data.

***Index Terms*—Natural Language Processing, Low-resource learning, Active Learning, Online Learning, Prediction with expert advice**

I. Introduction

Recent approaches to several NLP tasks have been dominated by the deep learning trend, which has one important drawback: most models need to be trained with large amounts of labeled data. Labeled datasets have become increasingly available, enabling the success of deep approaches, but they are costly to produce and remain unavailable for many languages.

Active Learning (AL) [1], [2] is one of the most popular learning frameworks that aims to train models with limited annotated resources. Under this framework, an instance (or a small batch of instances) is iteratively selected from a pool of unlabeled instances according to a certain criterion of informativeness (often referred to as *query strategy*) to be labeled by a human annotator and added to the training set. Hence, a better performance can be achieved with a clever selection of the instances to be annotated. Many query strategies have been proposed to date; however, as shown by Lowell *et al*, the performance of these strategies may vary depending on the setting (task, dataset, or model) [3]. This implies evaluating which strategy is the most appropriate on a large enough (annotated) test set, which goes against the very principle of minimizing the need for annotated data that motivates AL. In fact, in truly low-resource scenarios, finding the best strategy in this fashion simply might not be possible.

To address this shortcoming, we design the problem of dynamically converging to the best query strategy as a problem of *prediction with expert advice*. We propose a novel application of an online learning method that combines different AL query strategies with the goal of converging to the best ones regardless of the task in hand. Thus, unlike the traditional expert advice scenarios, in which the experts are typically classifiers whose goal is to predict labels or actions, our experts are the query strategies, which are in charge of choosing a batch of unlabeled instances for the human to annotate. The size of each expert's batch at each iteration varies according to its weight, which is adjusted according to a metric of how "useful" was the expert's previous selection of instances. To this end, we build on the well-known Exponentially Weighted Average Forecaster (EWAF) algorithm, which learns incrementally how to select the best expert. In our setting, the loss function for each expert and forecaster is computed with a heuristic, because we do not have access to the optimal outcome (which, in this case, would be the optimal set of instances to select).

In this paper, we evaluate our solution on sequence labeling tasks. In this family of tasks, the goal is to predict a label for each token of a sentence, considering that these labels might be dependent on each other and even reflect structural relations within the sentence. We validate our proposal in two distinct scenarios: Part-of-Speech tagging (using an English dataset) and Named Entity Recognition (using a Portuguese dataset). Specifically, we aim to address the following research questions:

1) *Does our expert-based solution converge to the best individual query strategy?*. In other words, can our solution converge to the best query strategy with EWAF when we compute the loss function with a heuristic?

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020 and the Air Force Office of Scientific Research under award number FA9550-19-1-0020. Vânia Mendonça is funded by an FCT grant with reference SFRH/BD/121443/2016.

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

2) *If so, how soon does it start to pay off?* In other words, how many instances must be queried by our solution in order to reach the performance of the best strategy?

The results for both tasks show that our expert-based solution does converge to the best strategy in each scenario, while using a heuristic to calculate the loss function within the EWAF algorithm. Moreover, the results suggest our solution is appropriate for a low-resource scenario: with an annotation budget set to half of the unlabeled data pool, it nearly reaches the performance of a version trained with the entire pool of data.

The remainder of this paper is organized as follows. In Section II, we present some background regarding Active Learning and Prediction with Expert Advice. In Section III, we describe our proposal. Section IV details the experiments that we have performed and Section V presents a discussion of the major findings. In Section VI, we present the related works and compare them to ours. Finally, in Section VII, we present the main conclusions and point to future work directions.

II. Background

A. Active Learning query strategies

The most common instantiation of AL is known as pool-based AL: a model learns from an initially small labeled set $\mathcal{L}$, which is iteratively augmented with an instance (or a small batch of instances) from a pool of unlabeled instances $\mathcal{U}$, chosen according to some heuristic of informativeness (*query strategy*) [2]. Query strategies can be grouped into *exploitation-based* strategies, *exploration-based* strategies, and strategies that combine both exploitation and exploration.

Exploitation-based strategies ground their selection on the model's decisions regarding how to classify the unlabeled instances. One of the most commonly used exploitation-based strategies is known as Uncertainty Sampling (US) [4]. This family of strategies selects the instances where the model is most uncertain about the labeling decision. The most direct uncertainty measure is known as Least Confidence (LC), and consists of choosing the instance whose labeling has the lowest posterior probability, as given by Eq.1 (where θ represents the parameters of the model).

$$\underset{x}{\operatorname{argmax}} = 1 - P_\theta(\hat{y}|x) \tag{1}$$

Since this measure does not account for the distribution of the remaining possible labelings, two other measures of uncertainty were proposed: Margin Sampling, which considers the difference between the probabilities of the most likely labeling and the second most likely labeling [5], and Entropy, which selects the instances with highest information entropy [6] across every possible labeling (or across the n most likely labelings). These measures are straightforward to apply and efficient as long as one can access the probabilities computed for each labeling in the prediction step.

Exploration-based strategies, on the other hand, focus on the representativeness of the instances. An example of this family is Exploration-Guided Active Learning (EGAL) [7], which selects the instances which are the least similar to the labeled set $\mathcal{L}$ (i.e., with greatest diversity from $\mathcal{L}$) and sorts them according to how similar they are to their neighbors in the remaining unlabeled set $\mathcal{U}$ (i.e. according to their density). The balance between diversity and density is given by a parameter w that varies between 0 (only diversity is taken into account) and 1 (only density is taken into account). One advantage of this family of strategies is that it does not require retraining the model, as long as one sets the strategy's batch size as the budget of instances to be sent to the human annotator.

Finally, a popular strategy that combines both exploitation and exploration is Information Density (IDen) [8]. This strategy selects the instances to query based on both how uncertain the model is and how similar they are to the remaining instances in $\mathcal{U}$, as given by (2).

$$\phi(x) \times den(x)^\beta \tag{2}$$

The first term of this product, $\phi(x)$, is the instance's uncertainty according to one of the measures previously described, and the second term, $den(x)$, is the average of the similarities between that instance and the remaining instances in $\mathcal{U}$, weighted by the exponent β.

Other query strategies have been proposed, mostly based on the model's current outcome or its expected improvement, but these are either less efficient than Uncertainty Sampling and/or tend to perform worse in sequence labeling tasks [8], [9]; henceforth, in this paper we will focus on one representative strategy for each family of strategies: US as the exploitation-based representative; EGAL as the Exploration-based representative, and IDen to represent the combination of exploration and exploitation.

B. Prediction with expert advice

Prediction problems with expert advice can be seen as an iterative game between a *forecaster* and the *environment*, in which the forecaster resorts to different sources (i.e., *experts*) to provide the best forecast [10]. At each round t, the forecaster F consults a set of weighted experts $\mathcal{E} = \{E_1, \ldots, E_K\}$ and has access to the predictions $f^t_{E_k}$ in the decision space $\mathcal{D}$ made by each expert $E_k \in \mathcal{E}$. Considering the experts' predictions, the forecaster makes its own prediction, $p^t_f \in \mathcal{D}$. At the same time, the environment reveals an outcome $y^t \in \mathcal{Y}$.

A popular online learning algorithm used in this scenario is the EWAF, a generalization of the Weighted Majority algorithm presented by Littlestone and Warmuth [11]. This algorithm has well-established performance guarantees, which include a bound on how fast it converges [10].

In EWAF, the prediction p^t_f made by the forecaster is given by:

$$p^t_f = \frac{\sum_{k=1}^{K} \omega_k^{t-1} f^t_{E_k}}{\sum_{k=1}^{K} \omega_k^{t-1}}. \tag{3}$$

At the end of each round, the forecaster and each of the experts incur a non-negative loss, ℓ^t_f and ℓ^t_k respectively, based on the outcome y^t revealed by the environment:

$$\ell_f^t, \ell_k^t : \mathcal{D} \times \mathcal{Y} \rightarrow \mathbb{R} \quad (4)$$

Then, the weights $\omega_1, \ldots, \omega_K$ of each expert $E_k \in \mathcal{E}$ are updated according to the loss incurred by each expert as shown in Eq. 5 (where η is a parameter of the algorithm).

$$\omega_k^t = \omega_k^{t-1} e^{-\eta \ell_k^t} \quad (5)$$

By setting η to:

$$\sqrt{8 \log |\mathcal{E}| / T} \quad (6)$$

it can be shown that:

$$\sum_{t=1}^{T} \ell_f^t - \min_{k=1,\ldots,K} \sum_{t=1}^{T} \ell_k^t \leq \sqrt{\frac{T}{2} \log |\mathcal{E}|} \quad (7)$$

thus ensuring that the forecaster quickly reaches a performance similar to that of the best expert [10].

In our setting, we assume that the forecaster does not have access to the environment's outcome, whereby the environment would have to present the optimal set of instances to select for the model. Hence, the forecaster learns its own loss and each expert's loss through a heuristic.

III. Active Learning as a problem of prediction with expert advice

How can we combine several AL query strategies as experts in order to dynamically reach the performance of the best strategy? First, we define our scenario as a problem of prediction with expert advice: each expert corresponds to a query strategy, and the decision space $\mathcal{D}$ corresponds to sets of unlabeled instances that are selected from a pool of unlabeled instances, $\mathcal{U}$ (note that in our sequence labeling scenario, each instance is a sentence). Our goal is to iteratively award a greater weight to an expert (i.e., a query strategy) that selects the most useful instances to be labeled by a human annotator. In particular, at each iteration t of our algorithm (illustrated in Alg. 1), each query strategy selects a small batch of the most informative unlabeled instances (line 7). Then, our forecaster selects a portion of each strategy's selected batch of instances which is proportional to that strategy's weight ω_k (line 8). The forecaster's batch is then delivered to a human annotator (line 9) and added to the labeled set $\mathcal{L}$ for the next iteration (lines 10, 15 and 16).

Here, in line with our low-resource motivation, we introduce a change to the traditional AL setup. Inspired by previous works such as [12], we attempt to reduce the annotation effort by presenting the human annotator with the labels predicted by the sequence labeling model for each selected instance. Thus, the annotator only needs to change the labels that were incorrectly predicted to the correct labels. Note that the annotator remains unaware of which instances were selected by each strategy.

Before moving to the next iteration, each strategy's weight needs to be updated based on how good was the selection decision of unlabeled instances. In this scenario, recall that we do not know what would be the outcome of the environment, i.e., the optimal set of instances to select for the human to annotate. This fact actually suggests that we could define our setting as a *multi-armed bandit problem* [10]. In this class of problems, the environment's outcome is also unknown at each iteration, but the forecaster learns its own loss for a selected action. In contrast with our setting, we assume that the forecaster does not learn its own loss directly from the environment but has to compute it through a heuristic. In addition, we use the same heuristic to compute the loss of every expert, which led us to use EWAF's weight update rule, as previously shown in (5). However, one could also slightly change the previous setting to frame it as a multi-armed bandit problem. In this new setting, the forecaster would only select the prediction of one expert and use the heuristic to calculate its own loss at each iteration. Then, only the weight of the selected expert would be updated, using algorithms for multi-armed bandit problems, such as the Exponential-weighting for Exploration and Exploitation (EXP3).

So how should we compute the loss l_k^t for each expert in our scenario, in the absence of the environment's outcome? To address this challenge, we propose to reward each query strategy based on a heuristic of how useful the instances selected should be. For each strategy, we define the EWAF loss as $\ell_k^t = -r_k^t$, where r_k^t is a reward that corresponds to the ratio between the number of predicted labels edited by the human annotator, num_edits_k, and the total number of tokens across all instances selected by that strategy within the forecaster batch, $total\_num\_tokens_k$ (line 11). Our reasoning is that, the greater the need for the human to edit the labels predicted for a given instance, the more useful such instance should be, i.e., our reward is a "proxy" of how these instances are expected to improve the model's performance. We then update each strategy's weights using the cumulative reward R_k^t (line 13).

IV. Experimental setup

A. Tasks and corpora

We validate our proposal in two distinct NLP tasks: Part of Speech (PoS) tagging and Named Entity Recognition (NER).

For PoS tagging, we used the Brown corpus [13][1], annotated with the Universal Tagset [14], which comprises 12 classes: `ADJ` (adjective), `ADP` (adposition), `ADV` (adverb), `CONJ` (conjunction), `DET` (determiner), `NOUN`, `NUM` (numeral), `PRT` (particle), `VERB`, `.` (punctuation) and `x` (words that do not fall in any of the previous categories, such as misspelled or abbreviated words). When pre-processing this corpus, we discarded the tokens tagged with the labels `.` (corresponding to punctuation) and `x` (corresponding to unknown words, such as misspelled or foreign words), as well as tokens containing digits and the special character `&` (often labeled as a conjunction).

As for NER, we used the Brazilian Portuguese corpus Paramopama [15] mixed with the European Portuguese corpus Second HAREM [16], whose tagset comprises five classes: `ORGANIZACAO` (organization), `PESSOA` (person), `TEMPO` (time), `LOCAL` (place), and `O` (for the remaining tokens). When

[1] Available under NLTK: http://www.nltk.org/nltk_data/

Algorithm 1 Active Learning with expert advice.

Input: labeled set $\mathcal{L}$, unlabeled set $\mathcal{U}$, $model$, experts $\mathcal{E}$, expert weights $\omega_1, \ldots, \omega_K$ expert batch size b_k, forecaster batch size b_f, $budget$

$\omega_1, \ldots, \omega_K \leftarrow 1$
for each $t \in budget$ **do**
  $total\_instances\_to\_add \leftarrow \emptyset$
  $model.train(\mathcal{L})$
  $model.predict(\mathcal{U})$
  for each $E_k \in \mathcal{E}$ **do**
    $instances\_selected_k \leftarrow E_k.select\_instances(\mathcal{U}, \mathcal{L}, b_k)$
    $instances\_to\_ask_k \leftarrow forecaster(instances\_selected_k, \omega_k, b_f)$
    $instances\_to\_add_k, num\_edits_k, total\_num\_tokens_k \leftarrow simulate\_human(instances\_to\_ask_k)$
    $total\_instances\_to\_add \leftarrow total\_instances\_to\_add \cup instances\_to\_add_k$
    $r_k \leftarrow \frac{num\_edits_k}{total\_num\_tokens_k}$
    $R_k \leftarrow R_k + r_k$
    $\omega_k^{t+1} \leftarrow e^{\eta R_k}$
  end for
  $\mathcal{L} \leftarrow \mathcal{L} \cup total\_instances\_to\_add$
  $\mathcal{U} \leftarrow \mathcal{U} - total\_instances\_to\_add$
end for

pre-processing this corpus, we discarded the tokens containing digits and special characters, such as punctuation.

B. Features

In line with the motivation that underlies our proposal, we mostly extracted features that would be trivial to obtain in a low-resource setting. Thus, for PoS tagging, we used the following set of features:

- the token to be classified;
- its previous and next n tokens (n = 1, 2, 3);
- the token's orthographic "suffix", i.e., its last c characters (c = 1, 2, 3), and its last p pronunciations, after applying a Grapheme to Phoneme (G2P) system[2] to the sentence (p = 1, 2, 3);
- the Named Entity tag to which the token belongs, after tagging the sentence with Stanford's NER system [17].
- the token length, represented by nominal categories: `small` (under 3 characters), `medium` (between 3 and 5 characters), and `large` (6 or more characters);
- the token frequency in the current set, represented by the quartile of the training/test set token distributions to which the token belongs.

As for NER, we used a subset of the features above: the token, the previous and next n tokens, and the orthographic suffixes. We also used three binary features that signal whether the current token, its previous token, and the next token start with a capital letter.

C. Implementation details

As our main classifier, we use a discriminative model estimated with linear-chain Conditional Random Fields (CRF), a well-established model for sequence prediction, proposed by Lafferty *et al* [18]. This model is appropriate in combination with potentially overlapping features, and the features do not need to fully specify a state or observation, making it expectable for a CRF model to be estimated from less training data. We used the CRFSuite implementation [19] with the default trainer, which learns using Gradient Descent and the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method. We set the trainer to include transitions that are possible, but not observed, and use the following parameters: coefficient for L1 penalty = 0.1; coefficient for L2 penalty = 0.01, and maximum number of iterations = 200.

As for the query strategies, we implemented one strategy from each family described in Section II-A:

- Uncertainty Sampling, using Least Confidence as the uncertainty measure[3] (**US-LC**);
- Exploration-Guided Active Learning, following [7]. We implemented this strategy with two different similarity measures: the Jaccard similarity between sentences [20] (**EGAL-Jac**), and the cosine distance of the contextual embedding of the sentences – we use pre-trained BERT embeddings[4] [21] as provided by "BERT as a service" [22] (**EGAL-BERT**). We set the parameter that balances diversity with density, w, to 0.5 (thus giving equal weight to both);
- Information Density, following [8]. We used the same uncertainty measure as US and the same similarity measures and w as EGAL (**IDen-Jac**, **IDen-BERT**). The density exponent β was set to 0.5.

[2]https://github.com/Kyubyong/g2p

[3]We did not use Margin Sampling nor Entropy due to their computational cost. Since the CRF we are using is provided as a black-box, we would need to compute the probabilities for all label permutations over the set of labels and each sentence's length instead of directly accessing the probabilities for the best labelings.

[4]For Portuguese, we use the multilingual pretrained embeddings.

We also included a baseline strategy that randomly samples the instances to ask the human (**Random**).

Finally, the human annotator was simulated using the gold annotations provided in the corpora used.

V. Experimental results

In this section, we report and discuss the performance results of our expert-based solution for each task, in comparison to each query strategy in isolation.

To compute the results, we started by shuffling each corpus and set aside 1000 sentences for the test set. We report the average performance of 10 runs over 10 shuffles of the remaining corpus (along with the confidence intervals for each iteration, using $p = 0.05$), from which we obtained an initial labeled set $\mathcal{L}$ with 5 sentences and an initial unlabeled pool $\mathcal{U}$ with 500 sentences.

In each pair task-language, we report the performances of:

- Each individual strategy listed in Section IV-C, using a batch size of 10 instances;
- Our expert-based solution, combining all the query strategies listed in Section IV-C, using a batch size of 10 instances (**Experts**);
- A fully supervised version, trained on all the instances in the unlabeled set, along with the labeled set (**Supervised**).

The performance of each version was measured at each iteration using F-Score (F_1) (as computed by `scikit-learn` [23]). The AL simulation lasted until a budget of up to 250 instances was reached.

Results for PoS tagging are shown in Fig. 1, and results for NER are shown in Fig. 2. For improved readability, the F_1 axis of each plot starts on 50%.

Does our expert-based solution converge to the best individual query strategies?

Since our scenario does not provide the environment's outcome, from which the predictor learns the loss functions, we do not know if EWAF's convergence guarantees hold. Thus, our first question is whether our solution is able to converge to the best strategies nonetheless.

For both tasks, we can see that the performance of both versions of our expert-based solution converges towards the performance of the best individual strategies (which are US-LC and IDen-BERT). Moreover, when the budget of instances to ask the human annotator is depleted, the performance of our experts' system nearly reaches the performance of the supervised system on the test set. For PoS tagging, our solution reaches F_1 = 89.91%, *versus* the F_1 = 91.21% obtained by the supervised version. As for NER, they perform even closer: the expert-based solution reaches F_1 = 90.68%, *versus* the F_1 = 91.46% obtained by the supervised version, which contains twice as many annotated instances than our annotation budget.

How soon does it start to pay off?

Our second question concerns our goal of performing low-resource NLP: specifically, we want to know how many instances do we need to query so that our solution reaches the performance of the best strategies. In order for our solution to be relevant in low-resource settings, it would be desirable to reach the best performance in few iterations.

For PoS tagging, the curve for our expert-based solution presents a similar evolution to that of the best individual strategies, but it is only after querying 120 instances that the difference between its F_1 and the best F_1 goes below 1%.

As for NER, most strategies exhibit a similar F_1 curve from the start. Our expert-based solution performs very closely to the best individual strategies, with the difference between its F_1 and the best F_1 starting below 1% and decreasing to 0.3% in the last iteration.

VI. Related work

A. Learning the query from data

A recent line of work in AL that addresses the challenge of adapting to different datasets consists of learning the query strategy from the data (usually referred as Learning To Active Learn (LTAL)). Instead of using a query heuristic like the ones described in Section II-A, LTAL approaches define the problem of finding which instances would be the best for the human to annotate as a learning problem: in addition to the main learner for the task at hand (e.g. the sequence labeler), there is a second model that, given an unlabeled pool of instances, outputs the instances that should be annotated.

This approach has been followed by several recent works in different NLP tasks (including sequence labeling) [24]–[27]. Wang *et al* propose the use of a query model in order to address the challenge of performing AL when using black-box models (for which it might not be possible to obtain information such as the model's confidence, which is crucial to Uncertainty Sampling strategies). They apply their proposal to the task of Semantic Role Labeling, evaluating it with different models [24]. Fang *et al* define the problem of learning the query as a reinforcement learning problem, and applied their approach to NER, outperforming a Uncertainty Sampling query [25]. Liu *et al* define the problem of learning the query as an imitation learning problem, and applied their approach to text classification and NER, outperforming [25] in the first task [26]. Vu *et al* build on the approach proposed by [26], but instead of learning the query from a higher-resource dataset, they learn it from the predictions of the main learner, using the high-resource dataset only for the initialization of the query model. This version outperformed both heuristic queries and previous LTAL approaches in text classification and NER ([25], [26]) [27].

At best, LTAL approaches assume that an appropriate transfer language, domain or task is available to initialize the model that is responsible for learning the query, which might not always be the case in low-resource scenario. Moreover, it requires the computational overhead of training a second model for the purpose of learning the query. Thus, instead of building on this approach, we hypothesized that the approach of combining the best choices of each strategy would be more aligned with our goal of performing NLP tasks in a low-resource setting.

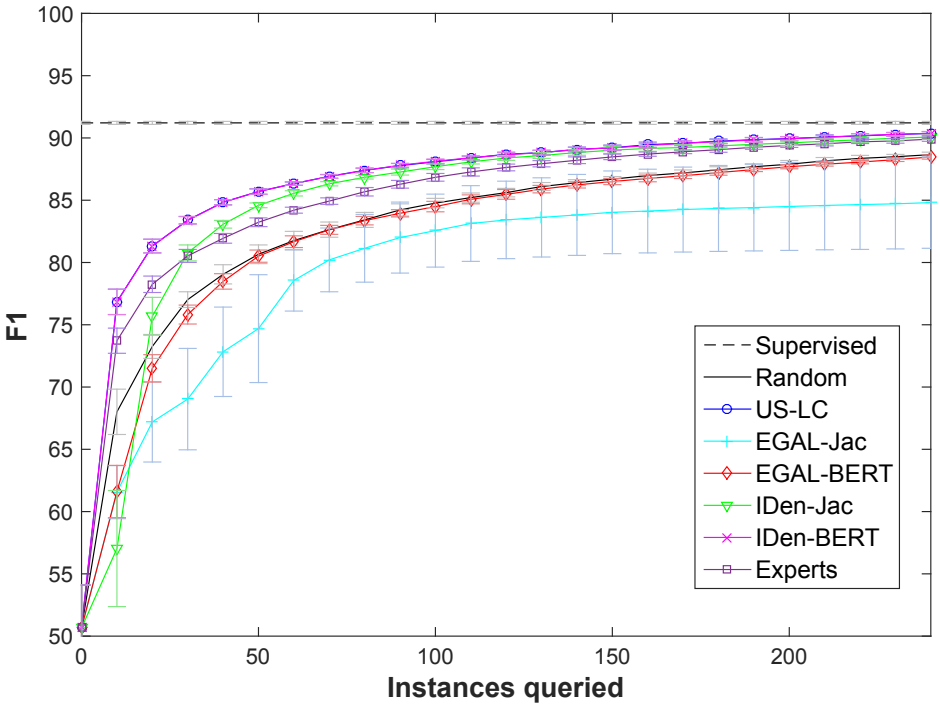

Fig. 1. F_1 results for PoS tagging on the test set.

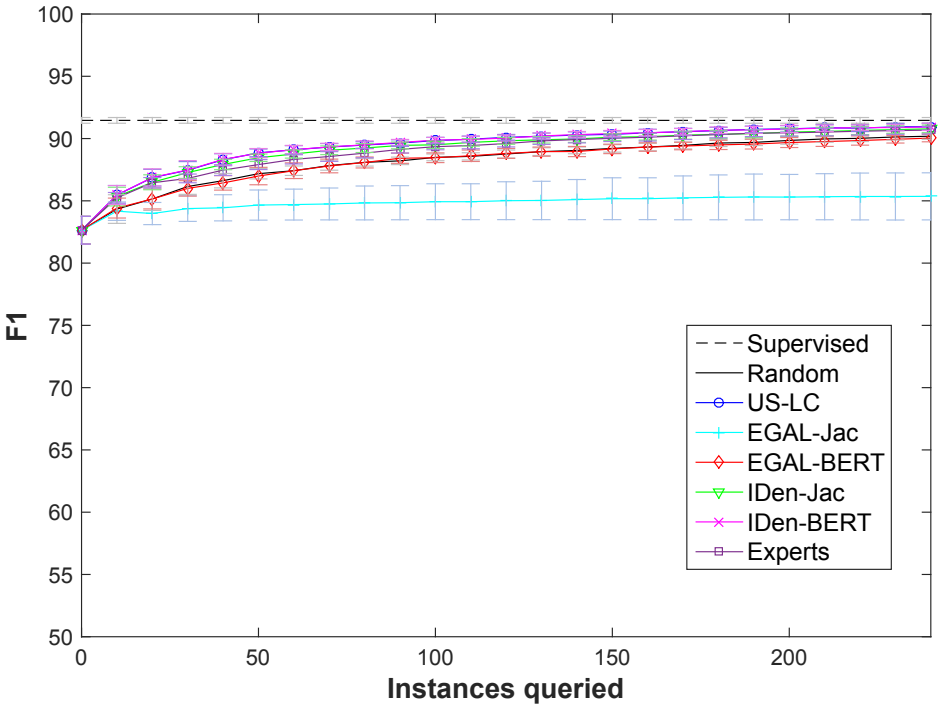

Fig. 2. F_1 results for NER on the test set.

However, it should be noted that LTAL could take part in our approach as one of the query strategies available.

B. Active Learning and expert advice

The combination of AL and prediction with expert advice has previously been applied in other works [28], [29], although only for binary classification problems.

Baram *et al* present an approach that, on the surface, is similar to the one we propose in this paper, as the environment's outcome is also unknown to the forecaster. However, their work displays some key differences. First, they defined their scenario as a multi-armed bandit problem, thus using the algorithms EXP3 and Exponential-weighting for Exploration and Exploitation with Experts (EXP4). When using EXP3, only one query strategy is chosen at each iteration, while, when using EXP4, the query strategies are combined, but they output their ratings of informativeness for each unlabeled instance instead of directly selecting instances. Second, their work only combines three query strategies, all of them highly tied to the Support Vector Machine classifier. Third, the approach is only applied and tested in a binary classification problem. Finally, another key difference is that they compute the loss for each expert using Classification Entropy Maximization. In the reported results, their combination of query strategies as experts kept up with the results of the best strategy on each problem [28].

Zhao *et al* combine AL with prediction with expert advice in order to avoid having to consult the environment's outcome in every iteration. They show, both formally and through empirical experiments on nine datasets, that their adaptation of two online learning algorithms with experts (EWAF and Greedy Forecaster) does not compromise the convergence guarantees of the original algorithms [29]. This work is significantly different from our approach because (i) the experts are linear classifiers that learn a binary classification task, and (ii) active learning is used only to reduce the number of requests to the environment to obtain the outcome.

Hence, to the best of our knowledge, this is the first work to apply prediction with expert advice for NLP problems, whereby the experts are AL strategies, and the experts' loss is computed in a heuristic manner.

VII. Conclusions and future work

In this work, we focused on the problem of performing NLP tasks in low-resource settings, using Active Learning. We addressed one important shortcoming of previous Active Learning approaches: the inconsistent performance of different query strategies across different settings. To this end, we presented a novel application of an algorithm for prediction with expert advice, EWAF, to dynamically combine different query strategies to select the unlabeled instances to be labeled by a human annotator. Even though we are working under different assumptions than EWAF, our expert-based solution managed to converge towards the performance of the best individual strategies in two different tasks and languages.

For future work, we intend to compare our approach with that of multi-armed bandits algorithms, EXP3 and EXP4; we also aim to include a LTAL strategy among our experts, and validate our proposal under a broader range of settings, such as different models.

Acknowledgment

The authors would like to thank Soraia Meneses Alarcão for proof-reading this document.

References

[1] D. Cohn, L. Atlas, and R. Ladner, "Improving Generalization with Active Learning," *Machine Learning*, vol. 15, pp. 201–221, 1994.

[2] B. Settles, "Active Learning Literature Survey," Tech. Rep., 2010.

[3] D. Lowell, Z. C. Lipton, and B. C. Wallace, "Practical Obstacles to Deploying Active Learning," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 21–30.

[4] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, 1994, pp. 3–12.

[5] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2189, 2001, pp. 309–318.

[6] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[7] R. Hu, S. J. Delany, and B. M. Namee, "EGAL: Exploration Guided Active Learning for TCBR," in *Proceedings of ICCBR*, 2010, pp. 156–170.

[8] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 1070–1079.

[9] V. Claveau and E. Kijak, "Strategies to select examples for Active Learning with Conditional Random Fields," in *CICLing 2017 - 18th International Conference on Computational Linguistics and Intelligent Text Processing*, 2017, pp. 1–14.

[10] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning and Games*. Cambridge University Press, 2006.

[11] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," *Inf. Comput.*, vol. 108, no. 2, pp. 212–261, 1994. [Online]. Available: http://dx.doi.org/10.1006/inco.1994.1009

[12] A. Culotta and A. McCallum, "Reducing labeling effort for structured prediction tasks," *AAAI*, vol. 5, pp. 746–751, 2005.

[13] W. N. Francis, H. Kučera, and A. W. Mackie, *Frequency analysis of English usage: Lexicon and grammar*. Houghton Mifflin Harcourt (HMH), 1982.

[14] S. Petrov, D. Das, and R. McDonald, "A Universal Part-of-Speech Tagset," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, 2012, pp. 2089–2096.

[15] C. Mendonça Júnior, H. Macedo, T. Bispo, F. Santos, N. Silva, and L. Barbosa, "Paramopama: a Brazilian-Portuguese Corpus for Named Entity Recognition," in *Encontro Nac. de Int. Artificial e Computacional*, 2015.

[16] C. Freitas, C. Mota, D. Santos, H. G. Oliveira, and P. Carvalho, "Second HAREM: Advancing the State of the Art of Named Entity Recognition in Portuguese," in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010*, 2010.

[17] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, 2005, pp. 363–370. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.8904

[18] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*, 2001, pp. 282–289. [Online]. Available: http://dl.acm.org/citation.cfm?id=655813

[19] N. Okazaki, "Crfsuite: a fast implementation of conditional random fields (crfs)," 2007. [Online]. Available: http://www.chokkan.org/software/crfsuite/
[20] P. Jaccard, "The distribution of the flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.
[21] J. Devlin, M.-w. Chang, L. Kenton, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Tech. Rep., 2018.
[22] H. Xiao, "bert-as-service," https://github.com/hanxiao/bert-as-service, 2018.
[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, Bertrand Thirion, Olivier Grisel, M. Blondel, Peter Prettenhofer, Ron Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[24] C. Wang, L. Chiticariu, and Y. Li, "Active learning for black-box semantic role labeling with neural factors," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 2908–2914.
[25] M. Fang, Y. Li, and T. Cohn, "Learning how to Active Learn: A Deep Reinforcement Learning Approach," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 595–605.
[26] M. Liu, W. Buntine, and G. Haffari, "Learning how to actively learn: A deep imitation learning approach," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 1874–1883.
[27] T.-T. Vu, M. Liu, D. Phung, and G. Haffari, "Learning how to Active Learn by Dreaming," in *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 2019, pp. 4091–4101.
[28] Y. Baram, R. El-Yaniv, and K. Luz, "Online Choice of Active Learning Algorithms," *Journal of Machine Learning Research*, vol. 5, pp. 255–291, 2004.
[29] P. Zhao, S. C. H. Hoi, and J. Zhuang, "Active Learning with Expert Advice," in *Uncertainty in Artificial Intelligence: Proceedings of the Twenty-Ninth Conference UAI 2013*, 2013, pp. 704–713.