`PyTAIL`: An Open Source Tool for Interactive and Incremental Learning of NLP Models with Human in the Loop for Online Data

Shubhanshu Mishra*
shubhanshu.com
mishra@shubhanshu.com

Jana Diesner
University of Illinois at Urbana-Champaign
jdiesner@illinois.edu

Abstract

Online data streams make training machine learning models hard because of distribution shift and new patterns emerging over time. For natural language processing (NLP) tasks that utilize a collection of features based on lexicons and rules, it is important to adapt these features to the changing data. To address this challenge we introduce `PyTAIL`, a python library, which allows a human in the loop approach to actively train NLP models. `PyTAIL` enhances generic active learning, which only suggests new instances to label by also suggesting new features like rules and lexicons to label. Furthermore, `PyTAIL` is flexible enough for users to accept, reject, or update rules and lexicons as the model is being trained. Finally, we simulate the performance of `PyTAIL` on existing social media benchmark datasets for text classification. We compare various active learning strategies on these benchmarks. The model closes the gap with as few as 10% of the training data. Finally, we also highlight the importance of tracking evaluation metric on remaining data (which is not yet merged with active learning) alongside the test dataset. This highlights the effectiveness of the model in accurately annotating the remaining dataset, which is especially suitable for batch processing of large unlabelled corpora. `PyTAIL` will be open sourced and available at `https://github.com/socialmediaie/pytail`.

1 Introduction

Analysis of large scale natural language corpora often requires annotation of dataset in a given domain with pre-trained models. Generally, these models are pre-trained on a fixed training dataset which is often different from the domain of the dataset under consideration. This often leads to poor performance of the model on this new domain. One way to address this gap is to utilize domain adaptation (Sarawagi, 2008; Daumé III, 2007) to improve the model accuracy. However, efficient domain adaptation requires labeled training data from the new domain, which is costly to acquire. The problem gets compounded for social media data (Mishra et al., 2015, 2014; Mishra and Diesner, 2016, 2018), for which the vocabulary and language usage continuously evolve over time (Mishra et al., 2019, 2020b; Mishra and Mishra, 2019). Take the example of sentiment classification, where the ways of expressing the same opinion also change with time. For example, the opinion label of the phrase "you are just like *subject*", will depend on the general opinion about *"subject"* when the phrase was expressed. Similarly, many new words are coined on social media (Eisenstein, 2013; Gupta et al., 2010). This poses a challenge for maintaining these models retain their accuracy over time. In this work, we propose an approach to alleviate this issue by creating a system based on active human-in-the-loop learning which incrementally updates an existing classifier by requiring an user to provide few new examples from the new data. Traditionally, this setup, called active learning (Settles, 2009) only deals with suggesting new training examples to annotate. However, since many NLP models use feature based on existing rules or lexicons, with changing data characteristics it may be more desirable to also suggest rule and lexicon updates in the model. Our system `PyTAIL` (Python Text Analysis and Incremental Learning) addresses the issues highlighted here by allowing human-in-the-loop active learning systems to integrate new data points, rules, and lexicons. Our main contributions are as follows: (i) Introduce `PyTAIL`, an open source tool with an active learning workflow which uses new data, rules, and lexicons to continuously train NLP models. (ii) Introduce a social media text classification benchmark for active learning research. (iii) Introduce an evaluation setup on unconsumed

*Work done while at University of Illinois at Urbana-Champaign.

data in active learning to quantify how quickly a corpus can be fully annotated with a reasonable accuracy.

2 Incremental learning of models with human in the loop

In this section we describe `PyTAIL` (Python Text Analysis and Incremental Learning). `PyTAIL`'s goal is to enable efficient construction of training data using active learning, while supporting incremental learning of models using the most recent data. A description of `PyTAIL` workflow is shown in figure 1. `PyTAIL` is built with the following features in mind: (i) Low cost of continuous training data acquisition (ii) Incorporation of domain knowledge using lexicon and rules (iii) Efficient update of model using only the newly acquired training data.

Lexicons and Rules Traditional active learning systems usually rely on only using the data as an input. However, `PyTAIL`'s focus is on involving humans at multiple stages of the learning process. Hence, `PyTAIL` relies on data, a set of lexicons, and a set of rules. The lexicons are used for counting lexicon matches, e.g. positive or negative words from sentiment lexicons. Rules are arbitrary rules for generating features from the data, e.g. presence of a regular expression pattern. The lexicons are often used via a rule to count lexicon matches in the text. These lexicons and rules are used to help human annotators make better decisions on annotating the data and also help in the training of the model. Our rules are inspired from the Labeling function approach of Snorkel (Ratner et al., 2019), however, they differ as they are used as feature generator.

Overview As shown in figure 1, the user starts with a collection of artifacts in the Bootstrap Stage. This can include an pre-trained model, a small seed training dataset, existing rules, and lexicons. Next, the user introduces their unlabeled data stream from their domain of interest, e.g. social media corpora. The bootstrap artifacts are used to predict this data stream. These predictions are then fed to the query strategy (described below) to identify artifacts for the suggestion stage. The user can then accept, reject, update these suggestions or even introduce new suggestions. Next, the model is updated using updated artifacts such that the rules and lexicons are used for updating the model features and the annotated data is used for updating the model. Finally, `PyTAIL` shows continuous evaluation metrics which include metric on a test set, user accepted training set, and unobserved data stream. This process is repeated till a stopping criteria is met, e.g. the exhaustion of data stream or achieving reasonable evaluation score. `PyTAIL` supports two modes for training, one is human in the loop (HITL) mode, and another is simulation mode. The simulation model uses pre-defined heuristics to simulate human actions based on model prediction scores. The default model when applied to benchmark datasets is the simulation mode.

Human in the loop (HITL) mode In the HITL mode, `PyTAIL` uses the pre-trained model to suggest top K instances to the user. The user can sort the instances using the scoring criterion. In order to reduce the cognitive work of labeling an instance from scratch, the user is shown the model predictions (as well as the label probability). The user is only required to edit the labels if they disagree. Model predictions for all the unlabeled instances from the top suggestions are now used as gold labels and fed to the model during the update process (this is similar to self-supervision with the possibility of human intervention). The user is also shown the prominent features for that instance, the user can select these features and mark them as useful or useless. Lexicon matches with the annotations are also shown, along with prominent key phrases in the unlabeled data stream. The user can choose to update the lexicon with these new suggestions. Once the model update has happened, the user is provided feedback on the change in model evaluation on a held out data.

3 Benchmark for social media active learning

We introduce an active learning benchmark of 10 social media text classification datasets consisting of 200K posts. These datasets cover sentiment classification, abusive content identification, and uncertainty indication and is derived from the SocialMediaIE benchmark (Mishra, 2021, 2020a,b). The dataset is available at `https://zenodo.org/doi/10.5281/zenodo.7236429`.

3.1 Sentiment classification

For sentiment classification we use the same data as in (Mishra and Diesner, 2018). A description of these data is shown in table 1a. Clarin (Mozetič

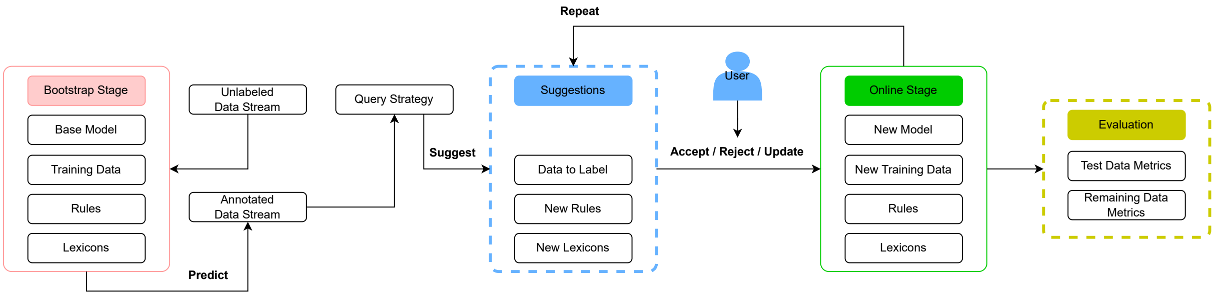


Figure 1: **`PyTAIL` Workflow**: Given a user and an unlabeled data stream, along with some bootstrapping artifacts, `PyTAIL` suggests data instances, rules, and lexicons which can be merged with bootstrapping artifacts to continuously create new model.

(a) Description of sentiment classification datasets. Datasets clustered together are enclosed between horizontal lines. Labels are *negative*, *neutral*, *positive*.

data	**split**	tokens	tweets	vocab
Airline	**dev**	20079	981	3273
	test	50777	2452	5630
	train	182040	8825	11697
Clarin	**dev**	80672	4934	15387
	test	205126	12334	31373
	train	732743	44399	84279
GOP	**dev**	16339	803	3610
	test	41226	2006	6541
	train	148358	7221	14342
Healthcare	**dev**	15797	724	3304
	test	16022	717	3471
	train	14923	690	3511
Obama	**dev**	3472	209	1118
	test	8816	522	2043
	train	31074	1877	4349
SemEval	**dev**	105108	4583	14468
	test	528234	23103	43812
	train	281468	12245	29673

(b) Description of abusive content classification datasets. Datasets which are clustered together are enclosed between horizontal lines. Labels for Founta are *abusive*, *hateful*, *normal*, and *spam*. Labels for WaseemSRW are *none*, *racism*, and *sexism*.

data	**split**	tokens	tweets	vocab
Founta	**dev**	102534	4663	22529
	test	256569	11657	44540
	train	922028	41961	118349
WaseemSRW	**dev**	25588	1464	5907
	test	64893	3659	10646
	train	234550	13172	23042

(c) Description of uncertainty indicators dataset. Datasets which are clustered together are enclosed between horizontal lines. Labels for Riloff are *sarcasm* and *not sarcasm*. Labels are for Swamy are *definitely no*, *definitely yes*, *probably no*, *probably yes*, and *uncertain*.

data	**split**	tokens	tweets	vocab
Riloff	**dev**	2126	145	1002
	test	5576	362	1986
	train	19652	1301	5090
Swamy	**dev**	1597	73	738
	test	3909	183	1259
	train	14026	655	2921

Table 1: Benchmark Datasets for Social Media Active Learning

et al., 2016) and SemEval are the two largest corpora. However, SemEval has a larger test set. All the sentiment datasets use the traditional labels of positive, neutral, and negative for labeling the tweets.

3.2 Abusive content classification

The second task we consider is abusive content classification. This task has recently gained prominence, owing to the the growth of abusive content on social media platforms. We utilize two datasets of abusive content. The first data is Founta from (Founta et al., 2018), which tags tweets as *abusive, hateful, normal, spam*. The second dataset is WaseemSRW from (Waseem and Hovy, 2016). It tags the data as *none, racism, sexism*. The rationale for including both these data under the same task it the core idea of identifing abusive content either direct or using racist or sexist variation. A description of these data is shown in table 1b.

3.3 Uncertainty indicators

Finally, we consider a collection of datasets for the task of identifying uncertainty indicators. Uncertainty indicators are defined as indicators in text which capture a level of uncertainty about the text, e.g., veridictality or sarcasm (uncertainty in intended meaning). We consider two datasets for this task as well. The first dataset is Riloff from (Riloff et al., 2013). This dataset consists of tweets annotated for sarcasm and non-sarcasm. The second dataset is Swamy from (Swamy et al., 2017). This dataset tries to identify the level of veridictality or degree of belief expressed in the tweet. The label set for this data is *definitely no, probably no, uncertain, probably yes, definitely yes*. A description of these data is shown in 1c.

4 `PyTAIL` for Social Media Text Classification

Model We use a logistic regression model with L_2 regularization. The regularization parameter is selected for each model using cross validation. We track the model scores on the held out test as well as validation data. Each text is represented using a set of features. Each tweet is tokenized and preprocessed by normalizing all mentions of hashtags, URLs, and mentions. We also use a large sentiment lexicon[1]. Furthermore, we suggest including a domain specific negative filter, i.e., words which should not be used to identify classification signals. For sentiment classification this can be entities in the corpora which should not bias the model.

Query selection strategies Active learning algorithms (Settles, 2009) identify most informative instances from unlabeled data that can be used to construct a high quality training dataset. The process of identifying informative instances is called **query selection**. Top instances $X_{selected}$ from the unlabeled data $X_{unlabeled}$ are identified based on a score. We consider two types of score: (i) $entropy = \sum_i p_i * log(p_i)$ - higher is better (ii) $min{-}margin = \max_{i \neq \star}\{p_i - p_\star \mid p_\star = \max_j p_j\}$ - lower is better. The entropy based scoring favors model predictions with highest randomness. The min-margin based scoring is useful in ensuring that the difference between the top prediction score and the second top prediction score is less. The selection is done using three strategies: (i) $Rand$: Instances are selected randomly without considering their scores, this acts as a baseline. (ii) X_{top}: Top K instances are selected based on their scores (X). (iii) X_{prop}: K instances are sampled proportional to their scores (X). This adds a degree of randomness to the top k strategy. These new instances are then added to the existing training instances $X_{train} = X_{train} \cup X_{selected}$, and the model is retrained.

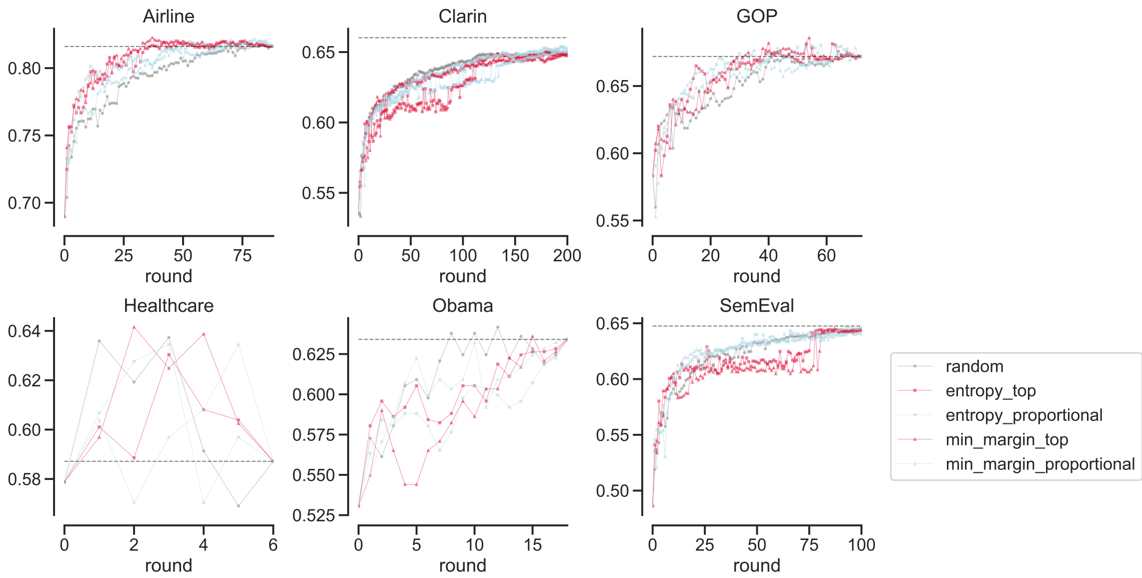
Evaluation on remaining dataset Active learning systems often just track the test dataset performance. However, we observe another dataset which is not used for training, it is the left over dataset X_{left} after selecting the examples in each round. X_{left} is continously decreasing and tracking the performance of the model on X_{left} can reveal how fast can an in-distribution dataset be accurately annotated using the specific querying strategy. This is suitable for simulation mode where the whole dataset ($X_{left} = X_{unlabeled}$) is already annotated.

Simulation Experiments Human annotation for `PyTAIL` can be simulated.First, X_train is set to $N = 100$ random samples from $X_{unlabeled}$. In each round, X_{select} is K (K=100) instances from $X_{unlabeled}$ based on the scoring criterion described above. We conduct 100 rounds of active learning (200 for Clarin as it is a very large dataset) and evaluate the models using the micro-f1 score. We also compare against a model trained on the full

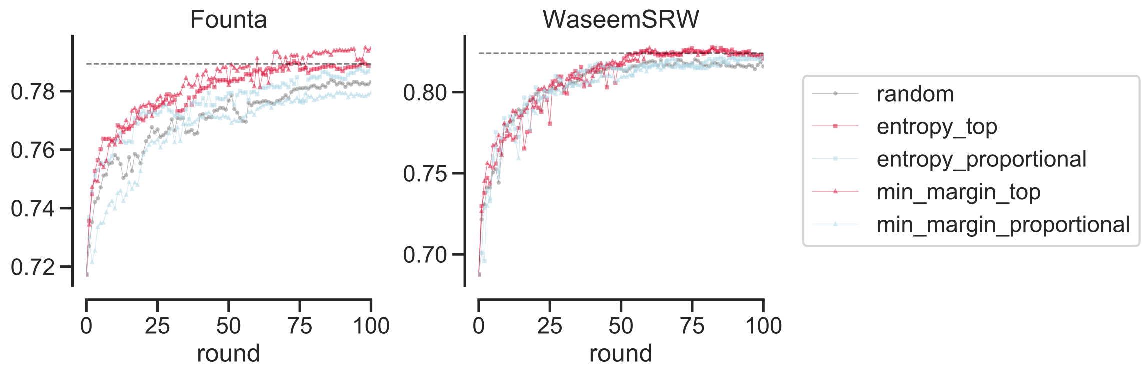
[1]https://github.com/juliasilge/tidytext/blob/master/data-raw/sentiments.csv

Table 2: Performance of query strategies across datasets using around 10% training dataset.

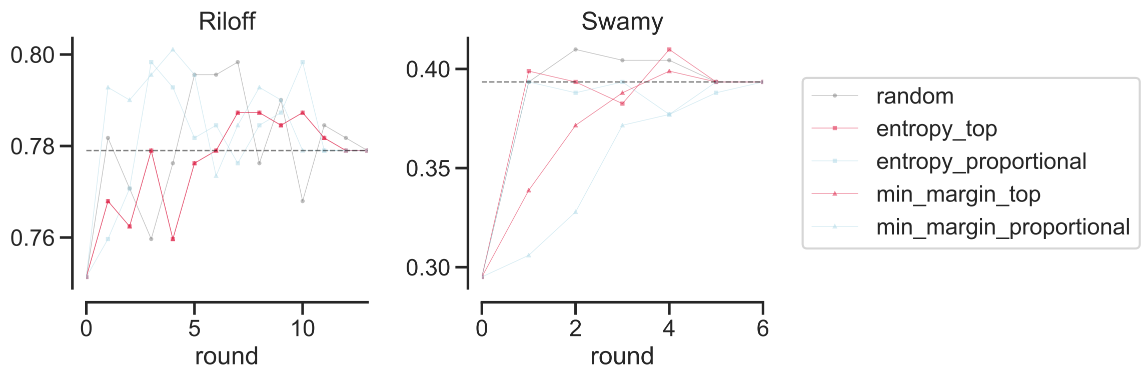
task	dataset	round	N	N_{left}	$\%_{used}$	Full	Rand	E_{top}	E_{prop}	M_{top}	M_{prop}
Test Dataset											
ABUSIVE	Founta	42	41,861	37,661	0.10	0.79	0.77	0.78	0.78	0.79	0.77
	WaseemSRW	14	13,072	11,672	0.11	0.82	0.79	0.78	0.77	0.78	0.76
SENTIMENT	Airline	9	8,725	7,825	0.10	0.82	0.76	0.78	0.79	0.77	0.77
	Clarin	45	44,299	39,799	0.10	0.66	0.63	0.61	0.62	0.63	0.63
	GOP	8	7,121	6,321	0.11	0.67	0.63	0.64	0.63	0.62	0.64
	Healthcare	1	590	490	0.17	0.59	0.64	0.60	0.61	0.60	0.60
	Obama	2	1,777	1,577	0.11	0.63	0.56	0.60	0.58	0.59	0.57
	SemEval	13	12,145	10,845	0.11	0.65	0.59	0.60	0.61	0.58	0.61
UNCERTAINITY	Riloff	2	1,201	1,001	0.17	0.78	0.77	0.76	0.77	0.76	0.79
	Swamy	1	555	455	0.18	0.39	0.39	0.40	0.39	0.34	0.31
Remaining Dataset											
ABUSIVE	Founta	42	41,861	37,661	0.10	NaN	0.77	0.80	0.78	0.81	0.78
	WaseemSRW	14	13,072	11,672	0.11	NaN	0.78	0.79	0.77	0.80	0.76
SENTIMENT	Airline	9	8,725	7,825	0.10	NaN	0.75	0.79	0.79	0.80	0.78
	Clarin	45	44,299	39,799	0.10	NaN	0.62	0.62	0.62	0.64	0.63
	GOP	8	7,121	6,321	0.11	NaN	0.62	0.64	0.62	0.63	0.63
	Healthcare	1	590	490	0.17	NaN	0.53	0.56	0.53	0.47	0.50
	Obama	2	1,777	1,577	0.11	NaN	0.54	0.56	0.57	0.56	0.56
	SemEval	13	12,145	10,845	0.11	NaN	0.61	0.62	0.62	0.63	0.62
UNCERTAINITY	Riloff	2	1,201	1,001	0.17	NaN	0.80	0.82	0.84	0.82	0.81
	Swamy	1	555	455	0.18	NaN	0.37	0.40	0.40	0.33	0.36



(a) Sentiment classification

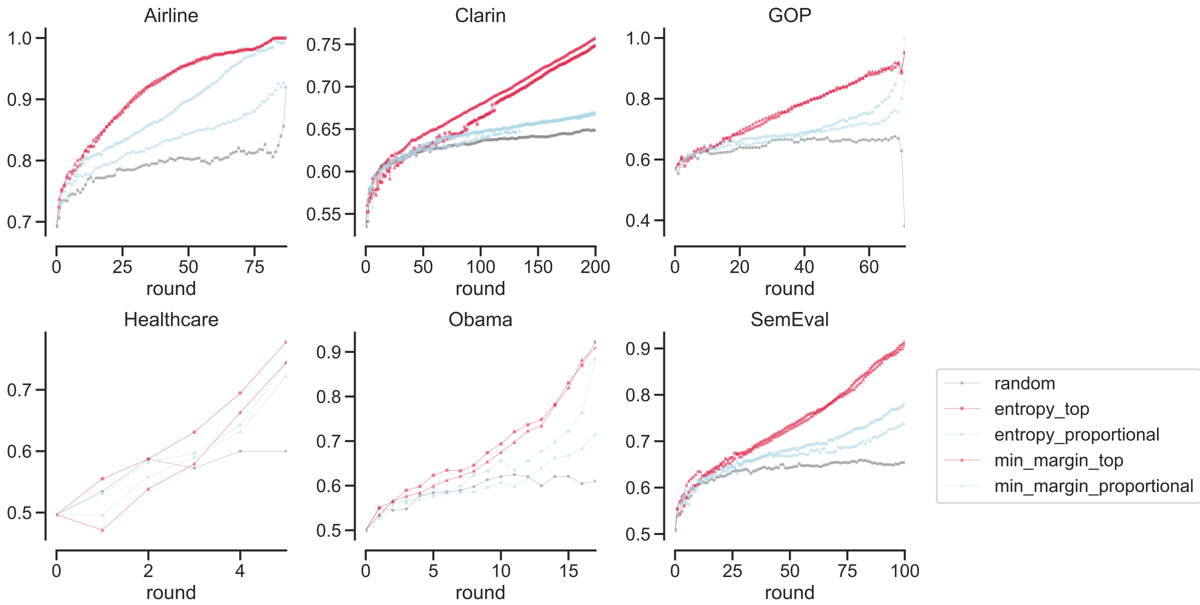


(b) Abusive content detection

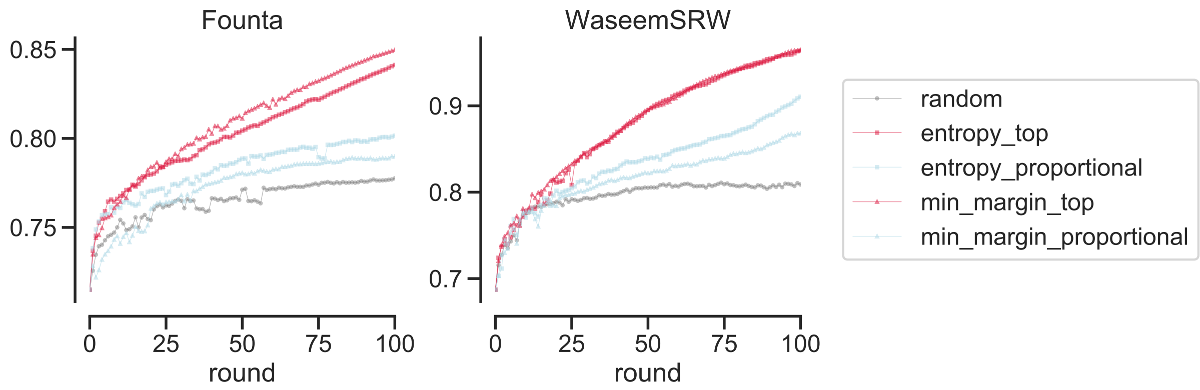


(c) Uncertainty indicators

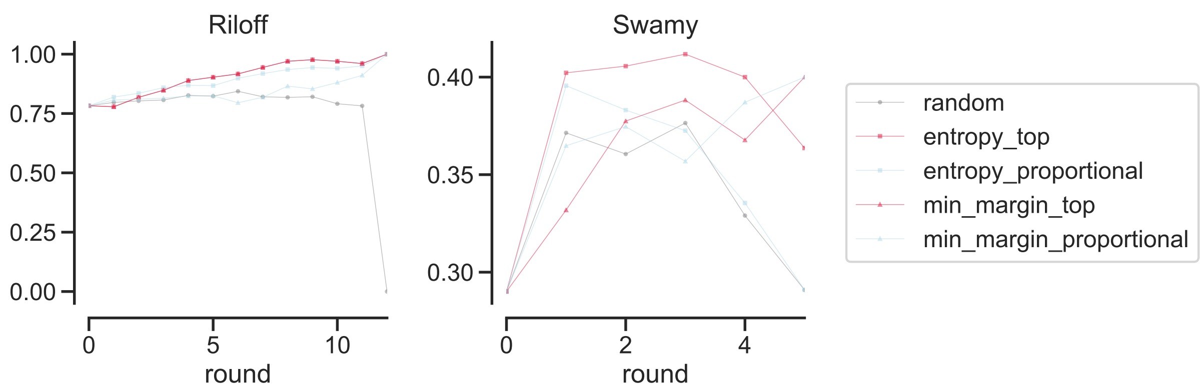
Figure 2: Progression of active learning classifier performance (micro f1-score) on the respective test set across 100 rounds of active learning (200 for Clarin). The annotation budget for each round is 100 instances, and the model is warm started with 100 random samples of the training data. Black dotted line is the classifier performance when trained on all of the training data. Data ordered alphabetically and X and Y axes are not shared.



(a) Sentiment classification



(b) Abusive content detection



(c) Uncertainty indicators

Figure 3: Progression of active learning classifier performance (micro f1-score) on the respective unselected data set across 100 rounds of active learning (200 for Clarin). The annotation budget for each round is 100 instances, and the model is warm started with 100 random samples of the training data. Data ordered alphabetically and X and Y axes are not shared.

data (Full). The experimental results on the test split of each data are shown in figure 2 and table 2. We observe that the top K strategy is usually the best followed by the proportional strategy across all data. For larger datasets we see that the model closes the gap very soon. We also show experimental results on the X_{left} part of the training data in figure 3. We observe that the top K strategy is consistently the best, followed by the proportional strategy across all data. The increase in performance on the X_{left} is indicative of the fact that active learning ensures that the remaining data is actually easy to annotate without human correction. This evaluation presents a more practical usage pattern of ML models. This usage pattern requires annotating pre-selected and large $X_{unlabeled}$. In reality, once the dataset is selected, one is interested in reducing the size of X_{train} to efficiently annotate the data. We think, it is in this setting that the active learning is most beneficial. If the user can achieve high labeling accuracy by annotating few samples, then the user's job is done.

5 Conclusion

We described experiments for evaluating active learning approaches for text classification tasks on tweet data. We introduced, `PyTAIL`, an open source user interface for active learning of NLP models by only requiring the user to update the labels for the model prediction if required. We also release a benchmark dataset for social media active learning. One limitation of our work is that our experiments are only conducted using simple linear model as they are easier to experiment with for sparse text features which we used for feature importance. However, the API does not place any restriction on the type of model.

In the future we plan to extend this strategy to non classification tasks for Social Media datasets e.g. structured prediction tasks like Named Entity Recognition, POS tagging, and Chunking (Mishra and Diesner, 2016; Mishra et al., 2020a; Eskander et al., 2022; Mishra and Haghighi, 2021; Mishra, 2019; Mishra et al., 2022; Mishra, 2020c).

`PyTAIL` is available as an open source tool at `https://github.com/socialmediaie/pytail/` and our dataset is available at `https://zenodo.org/doi/10.5281/zenodo.7236429`.

References

Hal Daumé III. 2007. Frustratingly Easy Domain Adaptation. *Association for Computational Linguistic (ACL)s*, (June):256–263.

Jacob Eisenstein. 2013. What to do about bad language on the internet.

Ramy Eskander, Shubhanshu Mishra, Sneha Mehta, Sofia Samaniego, and Aria Haghighi. 2022. Towards improved distantly supervised multilingual named-entity recognition for tweets. In *Proceedings of the The 2nd Workshop on Multi-lingual Representation Learning (MRL)*, pages 115–124, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. In *International AAAI Conference on Web and Social Media*.

Manish Gupta, Rui Li, Zhijun Yin, and Jiawei Han. 2010. Survey on social tagging techniques. *ACM SIGKDD Explorations Newsletter*, 12(1):58.

Shubhanshu Mishra. 2019. Multi-dataset-multi-task Neural Sequence Tagging for Information Extraction from Tweets. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media - HT '19*, pages 283–284, New York, New York, USA. ACM Press.

Shubhanshu Mishra. 2020a. Information Extraction from Digital Social Trace Data with Applications to Social Media and Scholarly Communication Data. *ACM SIGIR Forum*, 54(1).

Shubhanshu Mishra. 2020b. *Information extraction from digital social trace data with applications to social media and scholarly communication data*. Ph.d. dissertation, University of Illinois at Urbana-Champaign.

Shubhanshu Mishra. 2020c. Non-neural Structured Prediction for Event Detection from News in Indian Languages. In *Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation*, Hyderabad, India. CEUR Workshop Proceedings, CEUR-WS.org.

Shubhanshu Mishra. 2021. Information extraction from digital social trace data with applications to social media and scholarly communication data. *SIGWEB Newsl.*, 2021(Spring).

Shubhanshu Mishra, Sneha Agarwal, Jinlong Guo, Kirstin Phelps, Johna Picco, and Jana Diesner. 2014. Enthusiasm and support. In *Proceedings of the 2014 ACM conference on Web science - WebSci '14*, pages 261–262, New York, New York, USA. ACM Press.

Shubhanshu Mishra, Sneha Agarwal, Jinlong Guo, Kirstin Phelps, Johna Picco, and Jana Diesner. 2019. Tweet IDs annotated for enthusiasm and support towards social causes: CTE, cyberbullying, and LGBT.

Shubhanshu Mishra and Jana Diesner. 2016. Semi-supervised Named Entity Recognition in noisy-text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 203–212, Osaka, Japan. The COLING 2016 Organizing Committee.

Shubhanshu Mishra and Jana Diesner. 2018. Detecting the Correlation between Sentiment and User-level as well as Text-Level Meta-data from Benchmark Corpora. In *Proceedings of the 29th on Hypertext and Social Media - HT '18*, pages 2–10, New York, New York, USA. ACM Press.

Shubhanshu Mishra, Jana Diesner, Jason Byrne, and Elizabeth Surbeck. 2015. Sentiment Analysis with Incremental Human-in-the-Loop Learning and Lexical Resource Customization. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media - HT '15*, pages 323–325, New York, New York, USA. ACM Press.

Shubhanshu Mishra and Aria Haghighi. 2021. Improved Multilingual Language Model Pretraining for Social Media Text via Translation Pair Prediction. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 381–388, Online. Association for Computational Linguistics.

Shubhanshu Mishra, Sijun He, and Luca Belli. 2020a. Assessing Demographic Bias in Named Entity Recognition. In *Bias in Automatic Knowledge Graph Construction - A Workshop at AKBC 2020*.

Shubhanshu Mishra and Sudhanshu Mishra. 2019. 3Idiots at HASOC 2019: Fine-tuning Transformer Neural Networks for Hate Speech Identification in Indo-European Languages. In *Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation*.

Shubhanshu Mishra, Aman Saini, Raheleh Makki, Sneha Mehta, Aria Haghighi, and Ali Mollahosseini. 2022. Tweetnerd - end to end entity linking benchmark for tweets. In *Advances in Neural Information Processing Systems*, volume 35, pages 1419–1433. Curran Associates, Inc.

Sudhanshu Mishra, Shivangi Prasad, and Shubhanshu Mishra. 2020b. Multilingual Joint Fine-tuning of Transformer models for identifying Trolling,Aggression and Cyberbullying at TRAC 2020. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying (TRAC-2020)*.

Igor Mozetič, Miha Grčar, Jasmina Smailović, H Alani, Igor Mozetič, and A Scala. 2016. Multilingual Twitter Sentiment Classification: The Role of Human Annotators. *PLOS ONE*, 11(5):e0155036.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2019. Snorkel: rapid training data creation with weak supervision. *The VLDB Journal*, 29(2-3):709–730.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.

Sunita Sarawagi. 2008. Information extraction. *Foundation and Trends in Databases*, 1(3):261–377.

Burr Settles. 2009. Active Learning Literature Survey. Technical report, University of Wisconsin–Madison.

Sandesh Swamy, Alan Ritter, and Marie-Catherine de Marneffe. 2017. "i have a feeling trump will win..................": Forecasting Winners and Losers from User Predictions on Twitter. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1583–1592, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, Stroudsburg, PA, USA. Association for Computational Linguistics.