Small-text: Active Learning for Text Classification in Python

Christopher Schröder[†] CHRISTOPHER.SCHROEDER@UNI-LEIPZIG.DE

Lydia Müller[†§] LYDIA.MUELLER@UNI-LEIPZIG.DE

Andreas Niekler[†] ANDREAS.NIEKLER@UNI-LEIPZIG.DE

Martin Potthast[†] MARTIN.POTTHAST@UNI-LEIPZIG.DE

[†]*Leipzig University, Germany*

[§]*Institute for Applied Informatics (InfAI), Leipzig, Germany*

Abstract

We present `small-text`, a simple modular active learning library, which offers pool-based active learning for text classification in Python. It comes with various pre-implemented state-of-the-art query strategies, including some which can leverage the GPU. Clearly defined interfaces allow to combine a multitude of such query strategies with different classifiers, thereby facilitating a quick mix and match, and enabling a rapid development of both active learning experiments and applications. To make various classifiers accessible in a consistent way, it integrates several well-known machine learning libraries, namely, `scikit-learn`, `PyTorch`, and `huggingface transformers`—for which the latter integrations are available as optionally installable extensions. The library is available under the MIT License at `https://github.com/webis-de/small-text`.

Keywords: active learning, text classification, query strategies, transformers, Python

1. Introduction

Text classification—in the same way as most contemporary machine learning approaches—requires large amounts of data to achieve top performance. However, in most real-world use cases, labeled data does not exist and is expensive to obtain. Active learning (Lewis and Gale, 1994) solves this situation by repeatedly selecting unlabeled data, deemed to be informative according to a so-called query strategy. Subsequently, those samples are labeled by a human annotator, a new model is trained on all data labeled so far, and the process is repeated. Active learning aims at minimizing the amount of labeled data required, while maximizing the resulting model's performance, e.g., text classification accuracy.

Identifying and reusing code patterns is at the heart of software development, but also essential to science, especially in the fields of machine learning, where most experiments involve software to some extent. Moreover, in computer science, and especially in machine learning, demands for reproducible experiments are rising, which is why an increasing number of publications include a public code repository. However, many of these repositories reimplement existing approaches in their experiments. This is unfortunate, since, as described by Sonnenburg et al. (2007), there are many advantages of using existing software

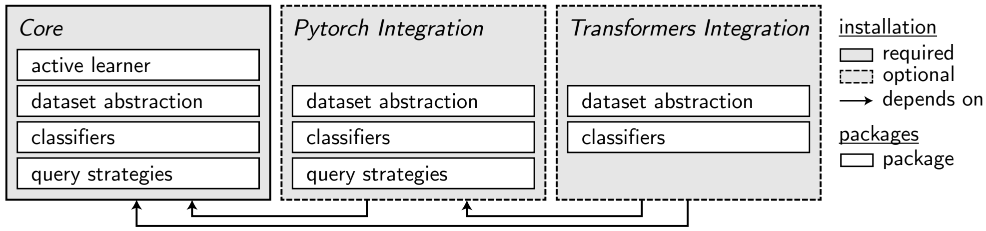

Figure 1: Modular architecture of `small-text`. The core installation can be extended by additionally installing a) the `PyTorch` integration, which allows to include GPU-based models, and b) the `transformers` integration, which allows to use state-of-the-art transformer-based sequence classification—all within an active learning setting. Dependencies between packages are omitted.

solutions in machine learning, among others, the faster detection of bugs, reduced implementation time, and improved reproducibility of experiment code. From a researcher's point of view the most important factor among those advantages is the reduced risk of having unspotted bugs—since a single unnoticed bug could invalidate the whole experiment. Second to that, the reduced effort of reusing existing components is critical and makes complex experiments possible which would otherwise have been infeasible. This particularly affects active learning, which typically consists of a classifier, a query strategy, and optionally a stopping criterion. Due to this modular nature, single components are mostly interchangeable, which results in many possible combinations and likewise in numerous lines of code. By implication, applied active learning necessarily involves a lot of code, and therefore it can benefit greatly from reusing existing open source software. Although there are existing solutions for general active learning, only few consider text classification, which requires additional functionality such as word embeddings (Mikolov et al., 2013) or languages models (Devlin et al., 2019). To fill this gap, we present a library, which provides robust components for building active learning experiments or applications for text classification.

2. Overview

The main goal of `small-text`[1] is to offer state-of-the-art active learning for text classification in a convenient and robust way for both research and applications. For this purpose, we built a modular pool-based active learning mechanism, exposing interfaces for classifiers and query strategies. The core of `small-text` integrates `scikit-learn` (Pedregosa et al., 2011), which makes it possible to directly use `scikit-learn`'s classifiers for active-learning. By providing the means to use many classifiers and query strategies within this setting, we can easily build active learning setups. We also implemented numerous query strategies, including such, which are only usable on text data (see Section 3). Furthermore, we provided integrations for two well-known libraries, namely `PyTorch` (Paszke et al., 2019) and `transformers` (Wolf et al., 2020), through which it is possible to easily use CUDA-based GPU computing and transformer models respectively. The architecture is again organized

1. Version 1.0.0a3 at the time of writing.

modular as shown in Figure 1, so that both integrations are completely optional, resulting in a slim core which can be used without unnecessary dependencies in a CPU-only scenario.

The library is available via the python packaging index and can be installed with just a single command via `pip install small-text`, and similarly, integrations can be enabled using the optional requirements arguments of Python's setuptools, e.g., the `transformers` integration is installed using `pip install small-text[transformers]`. The robustness of the implementation is secured by unit and integration tests. Detailed examples, an API documentation, and common usage patterns are available in the online documentation[2].

3. Query Strategies

The most important component of an active learning setup is the query strategy, which selects the instances to be labeled. We offer a wide range of query strategies: (i) confidence-based strategies: least confidence (Lewis and Gale, 1994), prediction entropy (Roy and McCallum, 2001), breaking ties (Luo et al., 2005); (ii) embedding-based strategies: BADGE (Ash et al., 2020), BERT K-Means (Yuan et al., 2020); and (iii) expected gradient length strategies: EGL (Settles et al., 2007), EGL-max (Zhang et al., 2017). Since there is an abundance of query strategies this list will likely never be exhaustive, and moreover, strategies from other domains such as computer vision are not always applicable to the text domain (e.g., Konyushkova et al., 2015).

4. Comparison to Previous Software

Unsurprisingly, with active learning having been researched and applied for decades, there are numerous other libraries available. For brevity's sake, we selected the most relevant —in the sense of projects having either a related publication or a larger user base—open-source projects, which are listed in Table 1 and discussed in the following.

Name	Language	License	Public Repository
`JCLAL`	Java	GPL	`https://github.com/ogreyesp/JCLAL`
`libact`	Python	BSD-2-Clause	`https://github.com/ntucllab/libact`
`ALiPy`	Python	BSD-3-Clause	`https://github.com/NUAA-AL/ALiPy`
`modAL`	Python	MIT	`https://github.com/modAL-python/modAL`
`lrtc`	Python	Apache 2.0	`https://github.com/IBM/low-resource-text-classification-framework`
`small-text`	Python	MIT	`https://github.com/webis-de/small-text`

Table 1: A descriptive overview of `small-text` and similar active learning libraries.

`JCLAL` (Reyes et al., 2016) is a generic framework for active learning which is implemented in Java and can be used either through XML configurations or directly from the code. Besides numerous single- and multi-label query strategies, it offers an experimental setting which, among others, shows a graphical report of the learning curve. The aim of `libact` (Yang et al., 2017) is to provide a convenient way to use active-learning in real-world ap-

2. `https://small-text.readthedocs.io`

plications. Notably, it includes a well-known meta-learning strategy (Hsu and Lin, 2015) as well as cost-sensitive active learning. Finally, `modAL` (Danka and Horvath, 2018) offers a broad range of active learning functionality, including regression, multi-label classification and stream-based active learning, as well as a multitude of query strategies. It also builds on `scikit-learn` by default, and provides instructions how to include GPU-based models using Keras and `PyTorch`. `ALiPy` (Tang et al., 2019) provides an active learning framework targeted at the experimental active learning setting. Apart from providing more than 20 query strategies, it supports extended active learning settings, e.g., active learning with noisy annotators. The `lrtc` (low resource text classification) framework (Ein-Dor et al., 2020) also focuses on text classification and has a number of built-in models, datasets, and query strategies to perform experiments on standard text classification benchmarks. It also provides a pre-implemented BERT classifier using the `transformers` library. Being focused on text classification, it is the most similar to our library, however, it is correctly called a framework, bringing an experimental frame, which is easily adaptable to other scenarios.

All of the listed libraries claim modularity and/or extensibility, however, this is partly owed to active learning's conceptual structure (i.e., components like classifiers, query strategies, and stopping criteria) in which the single components are usually independent as it is. While all of them have some overlap, only `modAL`, `lrtc` and `small-text` offer access to GPU-based deep learning frameworks, which is indispensable for text classification (and natural language processing in general as well) due to the recent success and ubiquity of transformer-based models (Vaswani et al., 2017; Devlin et al., 2019). Only `lrtc` and `small-text` focus on text classification, but they differ conceptually in so far that `lrtc` is a framework offering a set of procedures as a frame whose parts can be adapted, while `small-text`, a library, offers a tool box of components which can be used to build larger components or applications. Moreover, solely `small-text` offers the CNN-based gradient strategies of Zhang et al. (2017), which are unique to text classification and an important baseline for active-learning-based text classification. Most importantly, the transformers integration makes any transformers-compatible model usable in a very convenient way, thereby enabling state-of-the-art active learning for text classification in a few lines of code.

5. Conclusion

We introduced `small-text`, a modular Python library, which offers active learning for text classification. It integrates existing libraries like `scikit-learn`, `PyTorch`, and transformers, which makes state-of-the-art active learning easily accessible to the Python ecosystem. Due to its adherence to software engineering best practices, it provides a robust set of components to quickly apply active learning for text classification in both experiments and applications.

Acknowledgments

This research was partially funded by the Development Bank of Saxony (SAB) under project numbers 100335729 and 100400221.

References

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=ryghZJBKPS`.

Tivadar Danka and Peter Horvath. modAL: A modular active learning framework for Python. 2018. URL `https://github.com/modAL-python/modAL`. available on arXiv at `https://arxiv.org/abs/1805.00979`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.638. URL `https://www.aclweb.org/anthology/2020.emnlp-main.638`.

Wei-Ning Hsu and Hsuan-Tien Lin. Active learning by learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015. URL `https://ojs.aaai.org/index.php/AAAI/article/view/9597`.

Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Introducing geometry in active learning for image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.

Tong Luo, Kurt Kramer, Dmitry B. Goldgof, Lawrence O. Hall, Scott Samson, Andrew Remsen, and Thomas Hopkins. Active learning to recognize multiple types of plankton. *J. Mach. Learn. Res.*, 6:589–613, 2005.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *ICLR Workshop Track Proceedings*, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle,

A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL http://jmlr.org/papers/v12/pedregosa11a.html.

Oscar Reyes, Eduardo Pérez, María del Carmen Rodríguez-Hernández, Habib M. Fardoun, and Sebastián Ventura. Jclal: A java framework for active learning. *Journal of Machine Learning Research*, 17(95):1–5, 2016. URL http://jmlr.org/papers/v17/15-347.html.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML 01, pages 441–448. Morgan Kaufmann Publishers Inc., 2001.

Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1289–1296, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520. URL http://papers.nips.cc/paper/3252-multiple-instance-active-learning.pdf.

Sören Sonnenburg, Mikio L. Braun, Cheng Soon Ong, Samy Bengio, Leon Bottou, Geoffrey Holmes, Yann LeCun, Klaus-Robert Müller, Fernando Pereira, Carl Edward Rasmussen, Gunnar Rätsch, Bernhard Schölkopf, Alexander Smola, Pascal Vincent, Jason Weston, and Robert Williamson. The need for open source software in machine learning. *Journal of Machine Learning Research*, 8(81):2443–2466, 2007. URL http://jmlr.org/papers/v8/sonnenburg07a.html.

Ying-Peng Tang, Guo-Xiang Li, and Sheng-Jun Huang. ALiPy: Active learning in python. Technical report, Nanjing University of Aeronautics and Astronautics, January 2019. URL https://github.com/NUAA-AL/ALiPy. available as arXiv preprint https://arxiv.org/abs/1901.03802.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam

Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Yao-Yuan Yang, Shao-Chuan Lee, Yu-An Chung, Tung-En Wu, Si-An Chen, and Hsuan-Tien Lin. libact: Pool-based active learning in python. Technical report, National Taiwan University, October 2017. URL `https://github.com/ntucllab/libact`. available as arXiv preprint `https://arxiv.org/abs/1710.00379`.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.637. URL `https://www.aclweb.org/anthology/2020.emnlp-main.637`.

Ye Zhang, Matthew Lease, and Byron C. Wallace. Active discriminative text representation learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 3386–3392. AAAI Press, 2017.