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, an easy-to-use active learning library, which offers pool-based active learning for single- and multi-label text classification in Python. It features many pre-implemented state-of-the-art query strategies, including some that leverage the GPU. Standardized interfaces allow the combination of a variety of classifiers, query strategies, and stopping criteria, facilitating a quick mix and match, and enabling a rapid development of both active learning experiments and applications. To make various classifiers and query strategies accessible in a unified way, small-text integrates the well-known machine learning libraries scikit-learn, PyTorch, and huggingface transformers. The latter integrations are available as optionally installable extensions, making the availability of a GPU completely optional. The library is publicly available under the MIT License at https://github.com/webis-de/small-text, version 1.0.0b4 at the time of writing.

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

1. Introduction

Text classification—in the same way as most contemporary machine learning applications—requires large amounts of training data to achieve peak performance. However, in many real-world use cases, labeled data does not exist and is expensive to obtain—especially if domain expertise is required. Active learning (Lewis and Gale, 1994) solves this problem by repeatedly selecting unlabeled data deemed to be informative according to a so-called query strategy, which are then labeled by a human annotator. Subsequently, a new model is trained on all data labeled so far, and then this process is repeated until a stopping criterion has been met. Active learning aims at minimizing the amount of labeled data required, while maximizing the resulting model’s performance, e.g., in terms of classification accuracy.

An active learning solution therefore consists of corresponding components, namely a classifier, a query strategy, and an optional stopping criterion. Meanwhile, many strategies have been devised and studied for each of them. Seeing as choosing a worthwhile combination among them can only be done through extensive experimentation, and as implementing the different strategies is non-trivial while implementation details may differ dependent on the data at hand, this induces a considerable overhead. An obvious solution to this problem
is to employ open source libraries, which, among other benefits, accelerate research and ease
the adoption of methods by both researchers and practitioners (Sonnenburg et al., 2007). Although there are existing solutions for general active learning, only few consider active
learning for text classification, which requires functionality specific to the domain of natural
language processing, such as word embeddings (Mikolov et al., 2013) or language models
(Devlin et al., 2019). To fill this gap, we present a library, which provides tried-and-tested
components for active learning experiments with application to text classification.

2. Overview of Small-Text

The main goal of small-text is to offer state-of-the-art active learning for text classification
in a convenient and robust way for both researchers and practitioners. For this purpose,
we implement a modular pool-based active learning mechanism, illustrated in Figure 1,
which exposes interfaces for classifiers, query strategies, and stopping criteria. The core
of small-text integrates scikit-learn (Pedregosa et al., 2011), enabling direct use of its
classifiers. Overall, the library provides 13 query strategies, including some that are only
usable on text data, and two integrations of well-known machine learning libraries, namely
PyTorch (Paszke et al., 2019) and transformers (Wolf et al., 2020). The integrations
ease the use of CUDA-based GPU computing and transformer models, respectively. The
modular architecture renders both integrations completely optional, resulting in a slim core,
which can be used without unnecessary dependencies in a CPU-only scenario.

As the query strategy, which selects the instances to be labeled, is the most salient
component of an active learning setup, the range of alternative query strategies provided
covers four paradigms at the time of writing: (i) confidence-based strategies: least confi-
dence (Lewis and Gale, 1994; Culotta and McCallum, 2005), prediction entropy (Roy and
McCallum, 2001), breaking ties (Luo et al., 2005), and contrastive active learning (Marg-
gatina et al., 2021); (ii) embedding-based strategies: BADGE (Ash et al., 2020), BERT
k-means (Yuan et al., 2020), discriminative active learning (Gissin and Shalev-Shwartz,
2019), and SEALS (Coleman et al., 2020); (iii) gradient-based strategies: expected gradient
length (EGL; Settles et al., 2007), EGL-word (Zhang et al., 2017), and EGL-sm (Zhang
et al., 2017); and (iv) coreset strategies: greedy coreset (Sener and Savarese, 2018) and
lightweight coreset (Bachem et al., 2018). Since there is an abundance of query strategies,
Table 1: Comparison between small-text and relevant previous active learning libraries.

Unsurprisingly, after decades of research and development on active learning, numerous other libraries are available that focus on active learning as well. In the following we present a selection of the most relevant open-source projects for which either a related publication is available or a larger user base exists: 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. It offers an experimental setting which includes 18 query strategies. The aim of libact (Yang et al., 2017) is to provide active learning for real-world applications. Among 19 other strategies, it includes a well-known meta-learning strategy (Hsu and Lin, 2015). The modAL library (Danka and Horvath, 2018) offers active learning including regression, multi-label classification and stream-based active learning. It offers 12 query strategies, 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 this list will likely never be exhaustive—also because strategies from other domains, such as computer vision, are not always applicable to the text domain, e.g., when relying on the geometry of images (Konyushkova et al., 2015) and thus will be disregarded here.

The library is available via the python packaging index and can be installed with just a single command: pip install small-text. Similarly, the integrations can be enabled using the extra requirements argument of Python’s setuptools, e.g., the transformers integration is installed using pip install small-text[transformers]. The robustness of the implementation rests on extensive unit and integration tests. Detailed examples, an API documentation, and common usage patterns are available in the online documentation.

3. Comparison to Previous Software

Unsurprisingly, after decades of research and development on active learning, numerous other libraries are available that focus on active learning as well. In the following we present a selection of the most relevant open-source projects for which either a related publication is available or a larger user base exists: 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. It offers an experimental setting which includes 18 query strategies. The aim of libact (Yang et al., 2017) is to provide active learning for real-world applications. Among 19 other strategies, it includes a well-known meta-learning strategy (Hsu and Lin, 2015). The modAL library (Danka and Horvath, 2018) offers active learning including regression, multi-label classification and stream-based active learning. It offers 12 query strategies, 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

1. https://small-text.readthedocs.io
an active learning framework targeted at the experimental active learning setting. Apart from providing 22 query strategies, it supports alternative active learning settings, e.g., active learning with noisy annotators. The low-resource-text-classification-framework (lrtc; Ein-Dor et al. (2020)) also focuses on text classification and has a number of built-in models, datasets, and query strategies to perform active learning experiments.

In Table 1, we compare each of those projects and small-text by multiple criteria related to active learning or related to the specific code: While all libraries provide a selection of query strategies, not all libraries offer stopping criteria, which are crucial to reducing the total annotation effort and thus directly influence the efficiency of the active learning process (Vlachos, 2008; Laws and Schütze, 2008; Olsson and Tomanek, 2009). We can also see a difference in the number of provided query strategies, which is however not conclusive on its own: although a higher number of query strategies is certainly not a disadvantage, it is more important to provide the most relevant strategies (either due to recency, domain-specificity, or strong performance) since active learning experiments are computationally expensive (Margatina et al., 2021; Schröder et al., 2022) and therefore not every strategy can be tested during an experiment. Likewise, for application scenarios, only the one presumed best strategy will be employed. Only small-text and lrtc focus specifically on text classification, and solely modAL, lrtc and small-text offer access to GPU-based deep learning frameworks, which has become indispensable for competitive text classification due to the recent success and ubiquity of transformer-based models (Vaswani et al., 2017; Devlin et al., 2019). Finally, only small-text provides recent strategies such as BADGE (Ash et al., 2020), BERT K-Means (Yuan et al., 2020), and contrastive active learning (Margatina et al., 2021), as well as the gradient-based strategies by Zhang et al. (2017), where to latter are unique to active learning for text classification.

The distinguishing characteristic of small-text is the integration of scikit-learn, PyTorch, and transformers, which makes it possible to easily combine a wide range of classifiers, query strategies and stopping criteria. It provides a broad set of features including GPU support, stopping criteria, robustness through unit tests, and most importantly, it covers concepts that are specific to text classification such as embeddings, language models, and the text-based KimCNN (Kim, 2014). In summary, small-text offers a wide range of components, which are specifically targeted at text classification, thereby enabling state-of-the-art active learning for natural language processing using only a few lines of code.

4. Conclusion

We introduced small-text, a modular Python library, which offers active learning for text classification. It integrates scikit-learn, PyTorch, and transformers, and provides robust components to quickly apply active learning in both experiments and applications, thereby making state-of-the-art active learning easily accessible to the Python ecosystem.

Acknowledgments

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


