No Labels? No Problem! Experiments with active learning strategies for multi-class classification in imbalanced low-resource settings

Emiel Steegh
emielsteegh@gmail.com
University of Amsterdam
the Netherlands

Giovanni Sileno
g.sileno@uva.nl
Informatics Institute, University of Amsterdam
the Netherlands

ABSTRACT

Labeling textual corpora in their entirety is infeasible in most practical situations, yet it is a very common need today in public and private organizations. In contexts with large unlabeled datasets, active learning methods may reduce the manual labeling effort by selecting samples deemed more informative for the learning process. The paper elaborates on a method for multi-class classification based on state-of-the-art NLP active learning techniques, performing various experiments in low-resource and imbalanced settings. In particular, we refer to a dataset of Dutch legal documents constructed with two levels of imbalance; we study the performance of task-adapting a pre-trained Dutch language model, BERTje, and of using active learning to fine-tune the model to the task, testing several selection strategies. We find that, on the constructed datasets, an entropy-based strategy slightly improves the F1, precision, and recall convergence rates; and that the improvements are most pronounced in the severely imbalanced dataset. These results show promise for active learning in low-resource imbalanced domains but also leave space for further improvement.

ACM Reference Format:
Emiel Steegh and Giovanni Sileno. 2023. No Labels? No Problem! Experiments with active learning strategies for multi-class classification in imbalanced low-resource settings. In *Nineteenth International Conference on Artificial Intelligence and Law (ICAIL 2023), June 19–23, 2023, Braga, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3594536.3595171

1 INTRODUCTION

Many governmental organizations have an urgent need to regain control of their document archives [46]. In the Netherlands, for instance, the "Actieplan Open Overheid"[1], and international partnerships like the Open Government Partnership[2], dictate that government documents need to become publicly available; the "selection list"[3] indicates if and how long a document should be publicly available, based on the laws outlined in the *Archiefwet 2021*[4]. However, even if opening up governmental data is acknowledged to bring several benefits (e.g., transparency, enabling participation, increasing legitimacy), in practice, organizations are hesitant to act. Preparing data for publication is time-consuming [48], especially with unstructured data like text documents (e.g., permits, requests, complaints, notices, etc.), for which manual labeling is tedious and expensive [11, 14, 32]. From a broader perspective, it is easy to acknowledge that private organizations face similar challenges. With the exponential accumulation of digital assets, any administrative-oriented department has to tackle similar categorization problems.

Motivating problem. The essence of the problem can be described as such: A massive collection of (administrative) documents needs to be labeled according to some given categorization scheme, where there is a lack of quality labeled data. In general, such archives may contain videos, images, audio, emails, and various other types of documents; however, as textual data has a prominent role in administrative settings, this work focuses only on the problem of classifying textual documents. In particular, we are interested in contexts where minority languages are at stake (e.g., Dutch), texts may be generally written formally, and tend to include expert terminology (e.g., legal texts). This entails that at least part of the data belongs to a *low-resource* domain: it is neither English nor part of the standard training data of typical language models. Additionally, we assume categories to be highly *imbalanced*, as this scenario represents most practical cases.

Manually annotating an entire corpus is infeasible in these conditions; therefore, we aim to train an inferential model to assist with labeling. In principle, unsupervised (clustering) approaches are less suitable as they would not necessarily match the desired classificatory schema. Fortunately, when unlabeled data is abundant but manually acquiring labels is expensive, *active learning* is a relevant method to reduce the labeling burden [11, 43]. The technique (not confined to NLP tasks) aims to maximize the information gain of a machine learning model by drawing the most informative instances from the unlabeled pool. Even if active learning assists us in selecting the data points to manually label and use training, we still need to decide on a machine learning method to construct an inferential model for the task. For this purpose, large pre-trained language models (PTLMs) are commonly used in NLP today to achieve state-of-the-art results [39]. They provide language representations that can be used for downstream tasks in other datasets (e.g., text classification on a smaller dataset). In principle, PTLMs can be integrated

[1] https://www.rijksoverheid.nl/onderwerpen/digitale-overheid/open-overheid
[2] https://www.opengovpartnership.org/
[3] https://vng.nl/sites/default/files/2020-02/selectielijst_20200214.pdf
[4] https://wetten.overheid.nl/BWBR0007376/2022-05-01; https://www.nationaalarchief.nl/archiveren/kennisbank/wet-en-regelgeving

ICAIL '23, June 19–23, 2023, Braga, Portugal

ACM ISBN 979-8-4007-0197-9/23/06…$15.00
https://doi.org/10.1145/3594536.3595171

within an active learning process and transfer their embeddings to help learn the distribution underlying our classes, without the need to learn a general language first. Indeed, previous work has shown that active learning can successfully improve PTLM *accuracy convergence rates* (i.e., less training data is needed for the accuracy metric to approach the final score), even in an imbalanced binary classification setting [1, 11].

Although the introduction of active learning dates back over three decades, the earliest systematic study on employing active learning to fine-tune transformer-based PTLMs dates back only to three years ago [11], gaining traction in the context of transfer learning and large language models. Multiple active learning strategies have been explored in [27] to fine-tune a BERT model for classification on AG News [15], TREC-6 [38], and an internal Hindi-English hybrid dataset. The Mixed Aspect Sampling framework [1] proposes to use active learning to fine-tune a PTLM to detect misogynism and hate speech in text in imbalanced datasets. A more general active learning framework is investigated in [22], introducing a new sampling strategy and testing different strategies to fine-tune PTLMs to a host of common NLP benchmark tasks like topic classification, sentiment analysis, natural language inference, and paraphrase detection. However, all of these contributions are evaluated mainly on data sources similar to those used in PTLM training (e.g., English texts), with little imbalance and/or binary classes.

Classification on legal texts. Classification of legal texts (e.g., [3, 4, 33, 36]) is a field of its own. Although inspired by these works, we here approach the problem primarily from a low-resource, lack-of-quality labels perspective. Some of these papers, e.g., [3] find state-of-the-art results in PTLMs; others, e.g., [4] are critical of deep learning in the legal domain and show that it can be outperformed by concept-based machine learning approaches. Potential for greater results has been observed by integrating concept-based approaches with fine-tuned deep learning as in [35]. Since these works tend to operate in a fully labeled setting, active learning could be helpful in the complementary niche domain where labels are hard to come by.

Aims and contribution of the paper. Our research objective, aligned with the business problem outlined in the introduction, is to investigate multi-class classification in an imbalanced, low-resource setting. In this work, after constructing a dataset for this specific task, we run a series of experiments, creating the first insights on the effectiveness of active learning. Our main contributions are then: (1) We define and publish a dataset of (Dutch, legal) low-resource text for (imbalanced) multi-class classification task (section 3); (2) We show that active learning (section 4) can slightly outperform our random baseline in this task (section 5); and, more importantly, (3) we observe and discuss how active learning significantly boosts the convergence rate in a severely imbalanced dataset (section 6).

The code for this paper is available online[5].

[5]https://github.com/emielsteegh/mdwnlp

2 BACKGROUND

2.1 PTLM, Domain Adaptation, and Text Classification

General language understanding tasks like SuperGLUE [39] and its predecessor GLUE [40] are dominated by Pre-Trained Language Models (PTLMs). The shift from first-generation word embedding models to second-generation pre-trained contextual encoders can largely be attributed to the BERT model and the transformer architecture [9, 28]. They rapidly advanced accuracies on various tasks in the field. Extensive surveys and taxonomies have been presented eg. in [18, 28]. In downstream NLP tasks like text classification, PTLMs can transfer language representations [28] to reduce the labeling burden.

Domain Adaption. Different text domains have different language use [29]. Large language models like BERT usually learn general or domain-specific language representations during the pre-training based on their training data. In pre-training, it performs self-supervised language modeling tasks (e.g., next sentence prediction, masked language modeling) [18]. Training from scratch is the first step in training a PTLM [9]. SciBERT [2] is a domain specific PTLM trained from scratch on a corpus of science texts. Although proven effective, it is very time-consuming and requires a large volume of text. In *Continued Pre-Training* (CPT), parameters from a base PTLM (trained on a similar domain) are initialized, and the model continues pre-training on domain-specific data. For example, BioBERT[19] was created by initializing a base BERT and adapting it to Biomedical. Parameters are not learned from scratch; the base model parameters are tuned to the target domain. This requires fewer data than training from scratch if lower-level language concepts are similar between domains [18]. *Task-Adaptive Pre-Training* (TAPT) is an inexpensive way to adapt to a domain; the base model is initialized and performs only one or two epochs of continued pre-training on a small amount of unlabeled task-related data. Performing CPT or TAPT before fine-tuning can result in significant performance boosts in downstream tasks [10, 17, 18, 21, 23, 29], especially when dealing with a large volume of data without labels. In this work, we choose to perform TAPT and compare the performance of the adapted model to its base model. We choose BERTje [7] — a monolingual Dutch PTLM — as the base PTLM for our experiments because of its proximity to our target domain, due to the shared Dutch language.

Text Classification. In the supervised fine-tuning process for classification, the PTLM network is extended with one or more task-specific layers and then trained on a labeled dataset [28]. In low-resource text classification domains such as Dutch [9], German [30], Filipino [6], and legal [3], fine-tuned PTLMs consistently beat previous methods in terms of performance. Modifications can be made to the fine-tuning process to improve results further. We decide to follow [10, 22], which show that running multiple evaluations per epoch and keeping the checkpoint with the lowest validation loss consistently leads to better results.

2.2 Active Learning

After adapting a PTLM to the task domain, fine-tuning trains the model on the downstream classification task; however, supervised

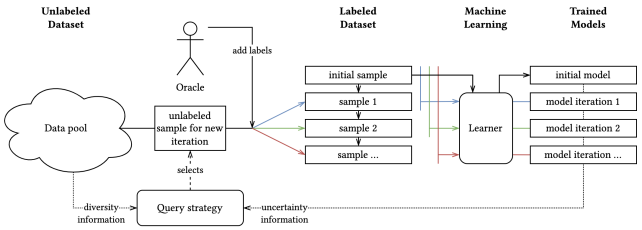

Figure 1: Overview of Active Learning: a Query Strategy $\mathcal{S}$ selects the most informative labels to be labeled by an oracle. The labeled instances are fed to a learning algorithm for training. $\mathcal{S}$ makes decisions based on information provided by the learned model and the data pool. The purpose is to make the model accuracy converge with less data.

classification needs labels. Active Learning (AL) is a technique that aims to improve model prediction accuracy at a lower cost by selecting the most informative instances for labeling [13, 32, 43]. Its purpose is to reduce the cost/time spent on human annotation. Figure 1 provides an illustration of AL. The Oracle O labels an initial sample of the unlabeled data $\mathcal{D}_{pool}$, creating the labeled data $\mathcal{D}_{lab}$. The model $\mathcal{M}$ is trained on $\mathcal{D}_{lab}$, producing the initial model. Then the active learning loop starts: The Query Strategy $\mathcal{S}$ selects a new sample from $\mathcal{D}_{pool}$ that is labeled by the Oracle and added to $\mathcal{D}_{lab}$, and then the model trains again on the new $\mathcal{D}_{lab}$. This loop repeats until we meet a *stopping criterion* $\mathcal{SC}$. The stopping criterion is frequently defined as a labeling budget (e.g., 1000 samples).

Query Strategies. The most important factor amongst the hyperparameters (query strategy, query sample size, initial sample strategy or size, stopping criterion) is the query strategy $\mathcal{S}(\mathcal{D}_{pool})$ [1]. Strategies usually fall into two categories: *diversity sampling* or *uncertainty sampling* [31, 32]. Diversity sampling methods aim to select samples representative of $\mathcal{D}_{pool}$, or the data source, to exploit the differences between classes in the feature space. Uncertainty sampling takes advantage of the model's low confidence predictions; *entropy* is usually the best performing classical sampling strategy. Random sampling is generally used as a baseline strategy. Depending on the fine-tuning implementation, this iterative random approach leads to better [22] or similar[34] accuracies compared to sampling the entire budget for one iteration of training.

2.3 Active Learning for NLP

It is known that active learning can improve convergence rates in NLP tasks [1, 12, 22, 32]. The investigation of PTLMs in active learning as a way to deal with small datasets was identified in [31] as an open research question.

State of the art query strategies. Recent state-of-the-art NLP active learning works use more complex query strategies that combine model uncertainty and data diversity. In the *contrastive active learning* (CAL) framework [22], $\mathcal{S}$ selects samples based on the similarity between labeled and unlabeled documents (diversity); these neighboring samples are then ranked based on prediction certainty. In [22], CAL outperforms all other strategies, yet Entropy sampling results are always among the bests. Active Learning by Local Sensitivity and Hardness (ALLSH) [47] determines sample informativeness by creating augmented data regions around their samples (diversity) and checking them for diverging predictions (uncertainty). [47] shows that ALLSH consistently outperforms Entropy and CAL accuracies in NLP tasks (except in question answering) by about $0.2 - 2\%$. Unfortunately, we could not access its code nor reproduce it. The Mixed Aspect Sampling (MAS) [1] for binary classification relies on *committee query strategies* to sample different aspects of the dataset. MAS scores as the best of the strategies, although the gap with other historically well-performing strategies is only 1-2%; yet it provides a significant accuracy improvement compared to the random baseline.

Practical obstacles for active learning in NLP. Two main obstacles are identified in the literature. First, uncertainty estimates are generally inaccurate [31]: PTLMs do not naturally produce class uncertainty scores the way earlier models used in active learning do (e.g., support vector machines). This can be overcome by training the model to predict certainty as well as the usual outcomes [1]. A second obstacle is the robustness of $\mathcal{D}_{lab}$, when it is tuned to a given model with no regard for diversity; uncertainty-sampled labeled sets can perform worse than a randomly sampled set when used to fine-tune another model [20]. This is strategically problematic: labeled data is a stable asset, whereas language models exhibit rapid changes.

Research Gap. Investigating the effectiveness of these approaches in a low-resource setting, the present work combines the adjacent-domain BERTje model *with* and *without* task adaptive pre-training in an active learning pipeline. We vary the query strategy between *random sampling* (as a baseline), *entropy sampling* (because it is shown to have consistently good performance in almost all works covered), and *contrastive sampling* (as a reproducible hybrid method, providing more robustness). Additionally, we perform our experiments in an imbalanced domain.

3 DATASET

To avoid issues with archives that may not be accessible or publishable due to their privacy-sensitive nature, we devised a proxy task to test whether state-of-the-art methods are still effective in a low-resource and imbalanced setting. This section explains why and how we acquired the "Open data Rechtspraak NetherLands" (`ORNL`) dataset and generated two datasets from it, `ORNL8` and `ORNL26`, for multi-class text classification.

Requirements. Considering our research context, we decided the dataset must be *multi-class*, with at least ten classes. The data points must be *labeled* with a class so that we can validate our learning results. The text must be in a language *other than English*, e.g., in Dutch. The texts must also be *out-of-domain* for the language model, so they cannot be generic texts such as news or Wikipedia articles. The dataset should have *more than 60k samples*. We prefer diverse contents to diverse structures; we suspect learning structural and metadata representations is easier than high-level linguistic patterns. We also want the dataset to be *imbalanced*; however, this is not strictly necessary, as imbalance is easy to mimic. Lastly, the dataset must be *publicly accessible*. We want to ensure reproducibility and allow other researchers to use the same dataset. Current

popular and public Dutch multi-class classification datasets do not sufficiently represent the outlined properties. They frequently have too few classes, are in-domain (e.g., CoNLL-2002 [37]), or miss truth labels (e.g., CC100-dutch [5]).

Source data. The Open Data of rechtspraak.nl[6] satisfies all of the previous requirements. It is Dutch judiciary data that consists of (anonymized) judgments, European Case Law Identifiers (ECLI), and sources of jurisprudence. The raw data is publicly available, categorized into four main classes, and optionally into one of the 26 sub-classes.[7] The individual files are tag structured. The five tags relevant to our case are `dcterms:subject, inhoudsindicatie, uitspraak & conclusie` (topic, short content description, verdict, conclusion), and the *European Case Law Identifier* (ECLI) as a unique identifier.

Furthermore, we create two separate task-oriented datasets for training. The first dataset, `ORNL8`, is artificially turned into a more class-balanced set with eight classes. The second dataset, `ORNL26`, is more representative of the actual class diversity, with 26 classes. The latter risks introducing far more noise, but it is more reflective of the substantial imbalance in the original problem.

3.1 The Dataset

The detailed process to get from the raw data to a usable dataset is available on GitHub[8]. The raw dataset (as available on July 2022) contains 229172 rows. Figure 2 visually shows the label balance. The calculated token overlap with our PTLMs tokenizer is 38% (Jaccard similarity coefficient 24%). Compared to the 42% overlap of SciBERT [2], we are fairly confident that WordPiece tokenization [25] will be able to handle the vocabulary difference.

3.1.1 Lightly and Severely Imbalanced. During processing, we extracted two overlapping datasets. Initially, we want to validate active learning as a strategy for out-of-domain NLP. The class imbalance is reduced in the first processed dataset, while it is fully captured in the second. To reduce the size of the dataset to manageable proportions, we limit the maximum amount of samples in the dataset. So we end up with two variations of the `ORNL` dataset:

- **`ORNL8`**: a lightly imbalanced dataset with eight total classes. From `ORNL` all subtopics with at least 5000 samples are included up to a maximum of 30000 samples (randomly sampled) per subtopic, resulting in eight total classes.
- **`ORNL26`**: a severely imbalanced dataset with 26 total classes. All subtopics from `ORNL` are included as a class, but each subtopic has a maximum of 5000 samples.

Both datasets are pre-split into a stratified train, validation, and test set (80/10/10). We published them to the python `datasets` package[9]. This makes it easy for researchers to use the datasets for further research.

[6]https://www.rechtspraak.nl/Uitspraken/paginas/open-data.aspx
[7]The data is stored in one XML file per judgment, bundled by year from 1905 to 2022. It is updated monthly and, as of 2022-06-14, contains 2580352 files (16.4 GB). Many judgment files are empty because the case was not fit for publication.
[8]https://github.com/emielsteegh/mdwnlp/tree/main/ORNL
[9]https://huggingface.co/datasets/Rodekool/ornl8 and https://huggingface.co/datasets/Rodekool/ornl26

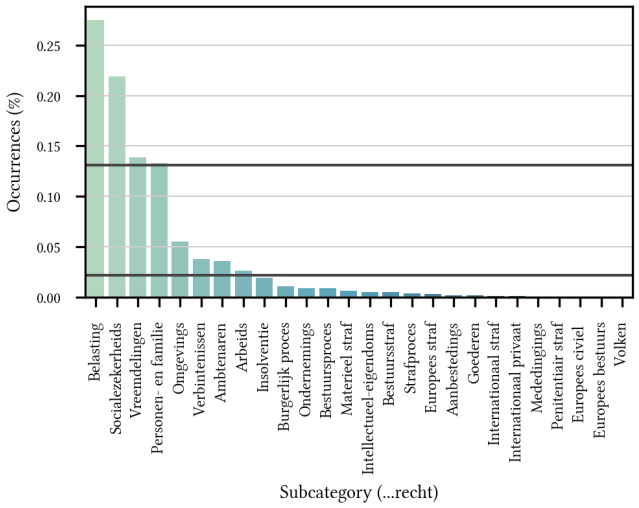

Figure 2: Occurrences of subtopics in the raw data. The upper horizontal line marks the maximum class size for `ORNL8` (30000). The lower horizontal line marks the minimum class size for `ORNL8` and the maximum class size for `ORNL26`.

4 METHOD

4.1 Task Adaptation

We use two variations of BERTje as our model $\mathcal{M}$. The first one is the off-the-shelf version `GroNLP/bert-base-dutch-cased`, without further pretraining. For the second model, we perform *task-adapted pretraining* (TAPT) and continue pretraining as recent literature suggests, using the implementation from the HuggingFace library [42]. This section outlines the TAPT process.

4.1.1 Performing Pretraining. Following the original training process of BERTje [7] we perform the same two tasks during continued pretraining: *Next Sentence Prediction* (NSP) and *Masked Language Modelling* (MLM) as described in [9]. In NSP, the model will need to predict whether sentences are successive. For the MLM part, the model must predict the masked token in a sentence. We prepare a dataset for continual fine-tuning: the dataset will have successive and non-successive sentence pairs; 15% of the tokens in the text will be masked.

Within the scope of this work, it was impossible to use the full datasets, as continued pretraining would take multiple days on a single Tesla V100.[10] Instead, we limited the pre-training data to 50000 documents sampled randomly. The texts are divided into paragraphs (1018033 paragraphs), and split into sentences by interpunction patterns (1965610 sentences).

4.1.2 Next Sentence Prediction. We create a dataset with three features: *sentence_a*, *sentence_b*, and a binary *next_sentence_label* (`isNextSentence` or `notNextSentence` in a 50/50 split). For NSP, this dataset will contain successive and non-successive sentence pairs. We create a new row in the dataset for all paragraphs with multiple sentences and assign any but the paragraph's last sentence

[10]We run the experiments on a single NVidia Tesla V100 or A100 (more GPU RAM is required for the 512 sequence length experiments) GPU.

Table 1: Table of model variations. Models marked with pt use the task-adapted (pre-trained) version of BERTje. The FS model uses all 50000 labeled documents for training.

ID	TAPT	$\mathcal{S}$	seq.len.
r-128 *(Baseline)*	-	Random	128
r-256	-	Random	256
pt-r-256	pt	Random	256
e-128	-	Entropy	128
e-256	-	Entropy	256
pt-e-128	pt	Entropy	128
pt-e-256	pt	Entropy	256
pt-e-512	pt	Entropy	512
cal-128	-	CAL	128
cal+256	-	CAL	256
pt-cal+256	pt	CAL	256
FS-256	-	Full Supervision	256

to *sentence_a*. In half of the cases, we add the next sentence to *sentence_b* and assign `isNextSentence`. In the other half, we assign a random sentence from all sentences (excluding the successive sentence) to `sentence_b` and set `notNextSentence`. Then we use the BERTje tokenizer to tokenize the sentence columns of this dataset with a sequence length of 512 tokens.

4.1.3 Masked Language Modelling. Following [9] we clone the current inputs of the tokenized dataset as our labels, then we replace 15% of the tokens in the dataset with a `[MASK]` token. A random float $p \in (0, 1)$ is assigned to all inputs in the dataset, each non-special token (`[CLS]`, `[SEP]`, padding) with $p \leq 0.15$ is replaced with the mask token.

4.1.4 Training. Finally, we write a separate dataloader for the TAPT dataset and run one epoch of continued pretraining through NSP and MLM on the model. With the *AdamW* optimizer and a learning rate of $5 \cdot 10^{-5}$.

4.2 Active Learning Implementation

Following [22], we implement and adapt the *Contrastive Active Learning Framework*. This allows us to run the experiments systematically with different parameters. We introduce new data loaders, models, and minor optimizations. For the core NLP, we use `BERTje` and `TAPT-BERTje` and attach a multi-class classification head specific to our dataset (through the HuggingFace Library [42]). Following [10], the model is evaluated five times per training epoch on the validation set, keeping the instance with the lowest validation loss. The models are evaluated against the evaluation and test splits described in section 3.

To ensure fairness and consistency between experiments, we perform each separate experiment (set of parameters) 5 times with the same predefined seeds ($\{672, 2451, 5262, 7763, 9105\} \in_R [0, 10000]$). The seed creates consistent pseudo-random sampled initial $\mathcal{D}_{lab}$, training set, and initialization of the head's feed-forward layer between experiments.

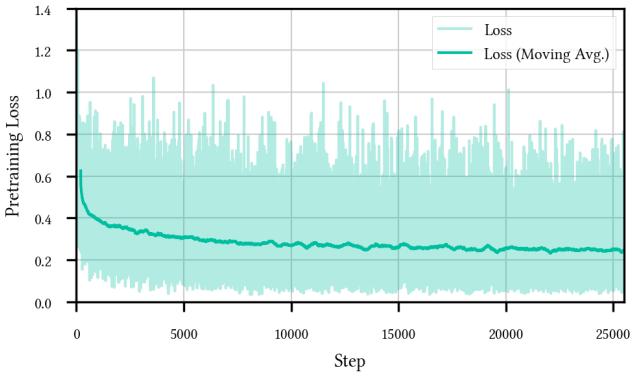

Figure 3: Loss Curve for Task-Adaptive Pre-Training.

We estimate a realistic labeling budget consisting of 3500 samples while still showing the effects of training the models. To reduce training times, we sample 50000 documents from the training pool. With a max pool size of 50000, we set the $\mathcal{SC}$ as a labeling budget to 7% of the $\mathcal{D}_{pool}$ (3500 samples). We handle an initial $\mathcal{D}_{lab}$ of 1% of the $\mathcal{D}_{pool}$ (500 samples) and then proceed to increase it by the same amount each iteration until the full budget is used.

The dataset includes labels, allowing us to mock the presence of an oracle. Instead of waiting for an expert to label the data manually, we can instantly assign the correct labels to the selected batch and continue the experiments.

The model variations used during our experiments are listed in Table 1. For the query strategy $\mathcal{S}$, we select *random* sampling as the baseline, *entropy* sampling as it is frequently cited as the best performing of the common strategies [1, 21, 22, 27] and *contrastive* sampling (CAL) as a more complex strategy because of its promising results in [22]. To establish a model performance "ceiling", we train a fully supervised model with the same basic parameters but using 100% of the data.

After running these experiments on only `ORNL8`, we compare the model performances early. We select the best-performing language model (`BERTje` or `TAPT-BERTje`) and run it on `ORNL26` with the AL strategies: random, entropy, and contrastive sampling. We eliminate trials with subpar performance on the simpler task, as we expect that their results will deteriorate further in more challenging conditions.

5 RESULTS

5.1 Task-Adaptive Pretraining

Figure 3 shows the cross-entropy loss during continued pre-training for task adaption. The curve is noisy, but the moving average has a downward trend and converges. A single round of evaluation of the NSP task on the `ORNL` test set shows that the No further Pre-Training (NPT) BERTje obtains 53.4% accuracy, and the Task Adapted Pre-Training (TAPT) BERTje obtains 90.3% accuracy, successfully improving on NSP.

5.2 `ORNL8` Results

Figure 4 summarizes the experimental results of models trained on the `ORNL8` dataset compared to the total quantity of documents

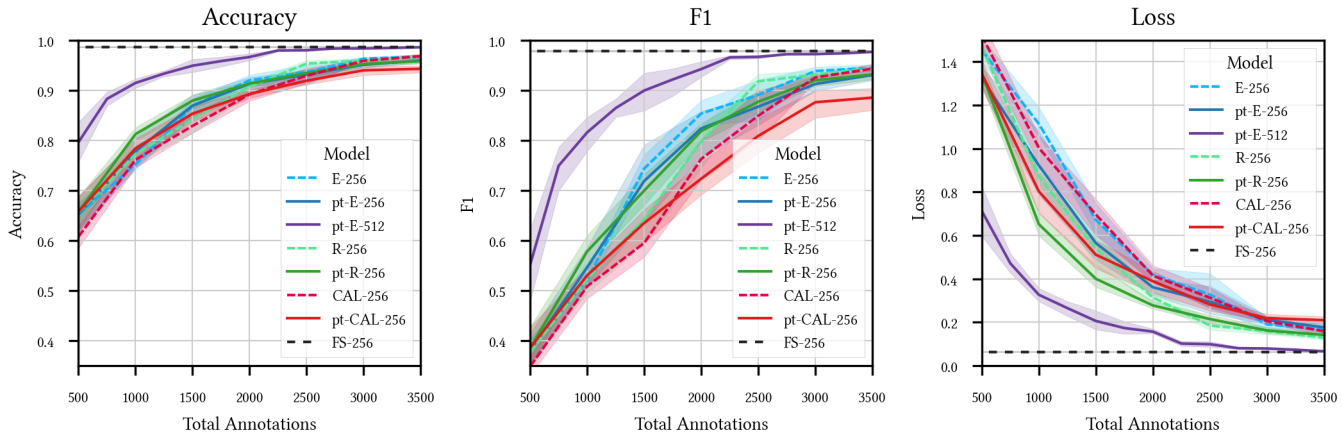

Figure 4: Accuracy, F1 and Loss for all models trained on ORNL8, excluding 128 sequence length models (dashed lines represent models without TAPT)

	Accuracy				*F1*				*Precision*		*Recall*		*Entropy*	
Model	mean	std	min	max	mean	std	min	max	mean	std	mean	std	mean	std
E-256	.9672	.0048	.9615	.9723	***.9448***	***.0080***	.9359	***.9536***	.9578	.0063	***.9373***	.0080	.6436	***.1166***
pt-E-256	.9590	.0065	.9517	.9665	.9300	.0127	.9162	.9452	.9453	.0084	.9208	.0148	.6074	.1234
pt-E-512	**.9857**	**.0004**	**.9850**	**.9863**	**.9763**	**.0008**	**.9754**	**.9776**	**.9763**	**.0013**	**.9763**	**.0008**	**.2252**	**.0250**
R-256	.9662	***.0027***	***.9622***	.9687	.9444	.0047	***.9383***	.9487	.9537	***.0056***	***.9373***	***.0053***	.3664	.1686
pt-R-256	.9580	.0060	.9481	.9634	.9314	.0122	.9107	.9419	.9354	.0067	.9290	.0156	***.2962***	.1386
CAL-256	***.9677***	.0065	.9581	***.9732***	.9428	.0127	.9234	***.9536***	***.9601***	***.0054***	.9323	.0152	.6094	.1578
pt-CAL-256	.9426	.0089	.9282	.9522	.8850	.0284	.8373	.9131	.9402	.0083	.8737	.0249	.6558	.1900
FS-256*	.9870	.0004	.9864	.9874	.9793	.0007	.9783	.9799	.9786	.0010	.9799	.0007	.0295	.0107

Table 2: Accuracy Metrics at 3500 Annotations on ORNL8. Per column best in bold, *second best in bold-italic*, Fully Supervised model (*) excluded.

annotated. All models perform very similarly on accuracy, except for the 512 token sequence length model. Table 2 details the accuracy and F1 scores after the labeling budget is exhausted: 3500 annotations.

5.2.1 Mean Accuracy. Model pt-E-512 outperforms the 256 token models by ≈ +15% at the start, reduced to a ≈ +1.5% at the end of the budget. It ends on par with the Fully Supervised 256-token model. Until around 1500 annotations, all TAPT models marginally outperform their NPT counterparts. Past 2000 annotations, all NPT models outperform their TAPT counterparts. All NPT active learning models beat their random counterparts by a fraction of a percent.

5.2.2 Mean F1. Again, pt-E-512 achieves better scores than the other models. Starting with a ≈ +16% higher F1, reduced to ≈ +4% as the entire budget is consumed. It ends on the same level as the FS model. The 256 token models end on a similar mean F1 score, except for pt-CAL-256's mean and standard deviation, the only outliers, to a negative extent.

5.2.3 Recall, Precision & Entropy. We observe that the models converge faster for precision than recall by about two iterations (a 1000 label difference). They all consistently achieve higher precision than recall. Surprisingly, CAL-256 performs highest on precision.

5.2.4 Accuracy vs. F1. At the first iteration, all models have substantially lower F1 scores than accuracies (≈ −24%), but as the models acquire more annotations, the difference decreases (to ≈ −1.5%). In Figure 5 we highlight this accuracy and F1 performance difference on the test set for pt-E-256 and pt-E-512.

5.3 ORNL26 Results

For ORNL26 Figure 6 shows the results of training the E-256, CAL-256 and R-256 models, as well as a fully supervised model (on a single seed) on the more imbalanced ORNL26 dataset. Table 3 shows the results at the full budget. We observe a far lower and more linear convergence rate in the models' performances here. The models approach the accuracy of FS-256 at a very similar rate but do not reach the score ceilings it sets. At budget, the F1 score of entropy sits at (≈ −20.6%) under full supervision. The entropy model has a higher mean F1 score (≈ +4.6%) but a lower mean accuracy (≈ −1.2%) than the random model. The models converge faster for

	Accuracy				F1				Precision		Recall		Entropy	
Model	mean	std	min	max	mean	std	min	max	mean	std	mean	std	mean	std
E-256	.8812	**.0088**	.8734	.8906	**.4450**	**.0110**	**.4324**	**.4520**	**.4886**	**.0194**	**.4498**	**.0042**	1.988	**0.6353**
R-256	**.8978**	.0117	**.8846**	**.9071**	.4094	.0188	.3902	.4279	.4260	.0258	.4226	.0190	**1.155**	1.2317
CAL-256	.8766	.0235	.8624	.9038	.3440	.0440	.3062	.3923	.3875	.0472	.3488	.0543	1.634	1.2683
FS-256*	.9581	-	.9581	.9581	.6496	-	.6496	.6496	.6679	-	.6496	-	0.172	0.1540

Table 3: Accuracy Metrics at 3500 Annotations on the More Imbalanced ORNL26. Per column best model in bold, Fully Supervised model (*) excluded.

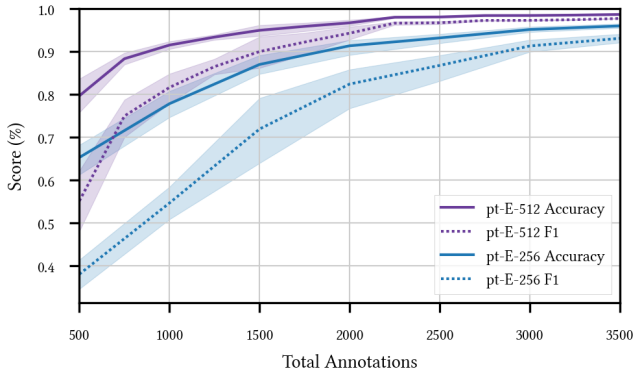

Figure 5: Comparison of Accuracy and F1. A bigger token input size increased both counts of performance, accuracy converges faster than F1.

precision than recall by about two iterations (or 1000 additional labels). They all consistently achieve higher precision than recall. In ORNL26 the E-256 outperforms both R-256 on both precision (≈ 14.6%) and recall (≈ 8.4%), while CAL-256 lags behind.

5.4 Training Time

Training times were dependent on the GPU used. The NVidia A100 would achieve up to 7 ± 0.8 iterations per second during training, whereas the NVidia Tesla V100 would run about 1.6 ± 0.5 seconds per iteration. Most experiments were performed on the V100 and would take to train one model, on average, 1 hour and 42 minutes with random sampling, 2 hours for Entropy active learning, and 2 hours and 36 minutes for Contrastive active learning. We focus now on three distinct phases of the active learning loop.

5.4.1 Training time. All models take the same time for training; it does not depend on the batch of samples selected as long as it is of the same size.

5.4.2 Inference time. This is the time spent on running the current model against the unlabeled $\mathcal{D}_{pool}$. The Random strategy needs no inference; Entropy and Contrastive $\mathcal{S}$ both make predictions for ≈ 309.8 ± 6.5 seconds per iteration. Both inference and selection time correlate with the size of $\mathcal{D}_{pool}$.

5.4.3 Selection time. This is the time $\mathcal{S}(\mathcal{D}_{pool})$ spends selecting the next batch. In Random and Entropy, this is almost instantaneous (random sample from $\mathcal{D}_{pool}$ and top k samples from the inference phase, respectively). Contrastive selection needs to compute for samples, taking ≈ 156.9 ± 35.9 seconds per iteration.

6 DISCUSSION

6.1 Task Adaption

The TAPT language model shows a significant improvement in next sentence prediction in the target domain compared to the original BERTje (subsection 5.1), and Figure 3 shows improvement in masked language modeling in the target domain. TAPT models have consistently lower entropy and loss scores than their counterparts.

In spite of these scores, the general performance of the TAPT model was worse. Although convergence rates (F1, accuracy, precision and recall) are marginally steeper (or the same) in the first two to three iterations of all TAPT variations of the models, the base PTLM consistently outperforms the TAPT model when there is more training data.

The results demonstrate that the language model was adapting to the new domain. BERTje's original training data does not include a large volume of legal text [7]. However, the comparable performance of the two versions on the downstream task may indicate that further training caused overfitting, that there was noise in the training data messing with the weights, or that the lower-level structures of the original training corpora and our legal texts are too similar. The best strategy to port BERT to different domains varies [3], especially in downstream tasks like classification. There might be no added value in TAPT before fine-tuning in our domain. It is possible that the PTLM will learn higher-level structures specific to our corpora from CPT, but the relatively small available dataset may lead to catastrophic forgetting [21, 23].

6.2 The ORNL8 Task

Performance on this task was generally good; at budget, all but two models scored between 95.9 – 96.7% accuracy and 93.0 – 94.5% F1 after labeling 3500 samples. The two exceptions: pt-CAL-256, which underperformed, and pt-E-512 which outperformed the rest, ending at 98.6% ± 0.04 accuracy (≈ +1.8% over the 2^{nd} best) and 97.7% ± 0.08 F1 (≈ +3.3% over the 2^{nd} best).

With a doubling sequence length, pt-E-512 gets more information per document. The model stabilizes at the scores of the fully supervised FS-256, which has only 256 token inputs. This raises questions of whether more tokens lead only to a higher convergence rate if they also lead to a higher performance ceiling, and if performance degrades like in [26]. We expect this effect is caused

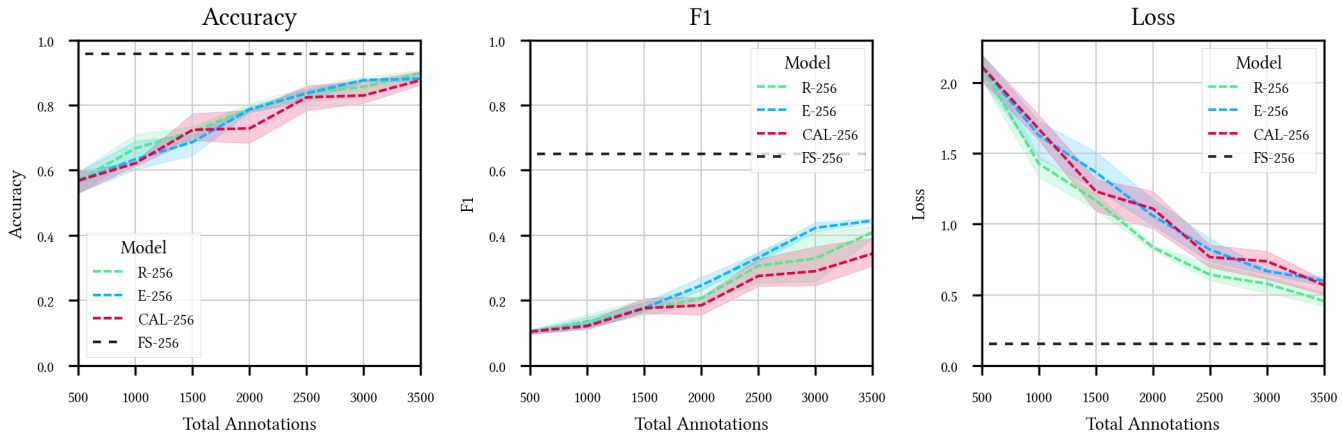

Figure 6: Accuracy, F1 and Loss for Models Trained on the imbalanced dataset ORNL26.

by how information that separates documents between categories is distributed in the text. Not all documents may give it their class away up front. The short content placed in front of a document is missing in about 8% of the texts.

The 128 token sequence length models served as an initial test and tuning of the system. However, they could not capture enough information in the texts, resulting in uninformative behavior. We replaced them with 256 token models early on.

6.3 The ORNL26 Task

The ORNL26 classification task (imbalanced datasets) is far more challenging for the models; the fully supervised model only manages a 64.9% F1, and the previous models perform significantly worse on this task. With 26 classes, of which 14 have less than a 1% presence, it is challenging to learn good representations of these classes without help. Entropy-based AL outperformed random sampling consistently, and Contrastive AL performed worse than the other two models.

Our investigation shows that entropy sampling acquires far more samples from the under-represented classes than random sampling. Figure 6, Table 3 reflect the impact of selecting a more balanced sample; while accuracies are similar, the convergence rates and mean scores for F1, recall, and precision of E-256 are best (≈ +3.6%, 6.3%, and 2.7% better than random respectively).

6.4 Added Value of Active Learning

Related literature shows that AL can improve model performance in the right scenario [22, 24], especially in less balanced settings; we observe a similar effect. In the imbalanced ORNL26, the best AL method (E-256) caused a faster convergence rate as well as a significant increase in the accuracy metrics at budget. Considering the relative homogeneity of the data, these scores may be seen as a lower bound for the benefit AL can provide.

However, previous works also show that not all datasets can benefit from AL [11, 22], and our experimental results imply that ORNL8 belongs to that category. Although we see marginal improvements with AL strategies in the dataset, they are not significant. This is likely because random sampling will deliver results as good as AL methods in tasks where the content is representative and noiseless enough [11]. In this task, Entropy sampling could select samples more effectively than random.

Figures 4, 6 show models had a higher loss during training than other models, likely because they sample documents that have uncertain predictions. Although the intention is to provide more information, the language model must be robust enough to capture the patterns that distinguish the "difficult" documents.

Compared to [22], contrastive AL underperformed in our task; perhaps the parameters were inadequately tuned to the problem. Verifying this requires further experimentation. Similar to [22], no approach consistently beats the others.

In this scenario, AL (especially entropy) had a small positive influence. Yet, depending on how expensive it is to label data, having to label 1% fewer data may be worth it. As soon as the model starts performing to the desired standard, it can aid the oracle with labeling by suggesting "easy" documents or providing assistive predictions.

Training a model takes computational time, and the AL loop takes additional time (see subsection 5.4). The scalability of AL depends on the size of the unlabeled dataset and the complexity of the acquisition function. Contrastive AL would take up to 8 additional minutes per iteration. The trade-off between more time spent training and faster labeling assistance is unique to the context of the problem.

It is far more valuable to employ AL in a niche with few labels available, where labeling is expensive. However, if labels come cheap or are already abundant, it is likely not worth it. Of course, setting up AL and running proxy tasks beforehand is also a time-consuming endeavor. Nevertheless, running a proxy experiment as close as possible to the real problem helps build an understanding of the real problem and what steps should be taken in the actual setting.

6.5 Limitations

6.5.1 No Comparisson. We observe decent scores on the ORNL8 task and successful AL on ORNL26. However, since this dataset has no precedent, we cannot compare the scores directly to other works.

6.5.2 Labels Live Longer. A set of labeled documents lasts longer than a trained model. Lowell et al. [20] demonstrate that documents sampled for the model's highest predicted information gain do not naturally generalize well to a new model. In some cases, they even lead to worse than randomly sampled scores. Suppose models are replaced every few years and labels stay valid. In that case, verifying that the actively sampled datasets provide more information to a new model than a randomly sampled dataset is crucial to the efficacy of the AL approach.

6.5.3 Pool Limit. Capping the training pool at 50k samples (section 4.2) led to "impossible classes" in the ORNL26 task. Because we set out to run many experiments in a limited time frame, we chose not to use the entire dataset; although this saved time, we sabotaged the model's training. This limitation had little influence in ORNL8. But, due to the imbalance in ORNL26, some experiment seeds led to a $\mathcal{D}_{pool}$ with very few or no documents from the extremely scarce classes (e.g., volkenrecht, burgerlijkbestuursrecht). This sampling makes learning these classes very difficult or even impossible without a good zero/few-shot approach (section 7). In a focused investigation or real-world setting, we recommend not limiting the size of the $\mathcal{D}_{pool}$.

6.5.4 Not the Perfect Model. We did not try to optimize for the most accurate model for the task; only a few (AL) hyperparameters were tested and tuned. We ran a limited set of experiments designed to investigate the value of AL. We can confidently say that model performance can be improved in both datasets. For example, we recommend comparing PTLMs first and then continuing with the most promising. Late in the experimental phase, we became aware of RobBERT [8], a robustly trained Dutch PTLM that outperforms BERTje on many Dutch benchmarks, especially in smaller datasets.

6.5.5 Oracle. The experiments assume a "perfect oracle" that will get the label right every time, and we do not account for oracle imperfections during the labeling phases. Furthermore, AL methods select uncertain samples for labeling. These samples are likely more challenging to label for a "human oracle", possibly resulting in more frequent mislabeling under uncertainty sampling than under random sampling. Qualitatively testing with a human oracle or a simulated imperfect labeling agent could lead to insights.

6.5.6 Sequence Length. Sequence length has a definite influence on performance. However, the maximum realistic sequence length depends on the available GPU RAM and the base PTLM training approach. It is possible to train in smaller batches, but this adversely affects training time. In our task, the 256 token input size, on average, covers roughly the first 10% of the document, which includes the very information-rich short content. In-document information distribution should be considered when selecting the tokenization approach.

6.5.7 Cold Start. The experiments all start with an initial batch of randomly sampled labeled documents. A cold start is not ideal; the first batch is not sampled for maximum information gain and has a large potential to be suboptimal. A better approach would be to perform unsupervised clustering first [31], or follow [34, 45], which approach the first batch dynamically and achieve good results.

7 CONCLUSION

In this paper, we have presented datasets and various methods relevant for a multi-class document classification task in Dutch language and legal domain (low-resource setting) at two levels of imbalance: light (ORNL8), and severe (ORNL26). Using random sampling and full supervision, we set an initial benchmark performance.

We conducted a series of experiments to evaluate the performance of various active learning strategies. Our empirical results and reflection show that active learning strategies provide no tangible benefit in the lightly imbalanced dataset. We significantly improved performance over the baseline by increasing the model's token sequence length (acc: $\approx -0.03\%$, f1: $\approx +3.6\%$, precision: $\approx +6.3\%$, recall: $\approx +2.7\%$). In the severely imbalanced dataset, entropy-based active learning performs significantly better as an acquisition strategy, where it boosted the convergence rates and final score over the baseline at the budget (acc: $\approx +0.9\%$, f1: $\approx +3.2\%$, precision: $\approx +2.3\%$, recall: $\approx +3.9\%$).

In addition to comparing active learning strategies, we observed that, contrary to the literature, and regardless of the promising MLM and NSP training results, task-adaptive pre-training of the PTLM was unable to improve results in the downstream classification task, as it resulted in similar or slightly worse performances. We consider that the PTLM training set and our dataset's lower-level language structures are almost identical [9], but more investigation is required.

We show that an active learning approach positively and significantly affects the performance and convergence rate of PTLMs in a low-resource imbalanced dataset, but we also noted that there is room for improvement on our tasks.

Future Work. These experiments set the first steps for multi-class imbalanced text classification in non-English languages. In the legal domain, a pertinent next step is involving experts for application and evaluation. Practically, there is much room to experiment with and fine-tune different active learning strategies (e.g., [1, 47]) and other (multi) language models on the task; A unified active learning framework for NLP. Research with public repositories like [16, 22] and Python packages like ModAL have started laying the groundwork but still lack ease of use, abstraction, or modern NLP capabilities.

There was no capacity in this work to evaluate the labeled datasets in other models; we urge researchers of active learning to test the robustness of the acquired datasets following [20].

Due to the observed effects of input sequence size, we propose investigating the accuracy of PTLMs in downstream tasks as a function of input sequence size, as [41] did for common neural networks before the popularization of PTLMs. Since most BERT-based models are trained with 512 input tokens, going past 512 tokens requires choosing a model with a larger feature space or training one from scratch.

Our findings, as well as those of [1, 11], are promising for the application of active learning for strongly imbalanced data. To further improve in imbalanced settings, we see a high potential in the fusion of active learning with classical zero- and few-shot learning and point to [4, 35, 44] as a starting point for using matching (e.g., on class descriptors) to improve learning for underrepresented classes.

REFERENCES

[1] Md Abul Bashar and Richi Nayak. 2021. Active Learning for Effectively Fine-Tuning Transfer Learning to Downstream Task. *ACM Transactions on Intelligent Systems and Technology* 12, 2 (Feb. 2021), 24:1–24:24.

[2] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. ACL, 3613–3618.

[3] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets Straight out of Law School. arXiv:2010.02559 [cs]

[4] Haihua Chen, Lei Wu, Jiangping Chen, Wei Lu, and Junhua Ding. 2022. A Comparative Study of Automated Legal Text Classification Using Random Forests and Deep Learning. *Information Processing & Management* 59, 2 (March 2022), 102798.

[5] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. arXiv:1911.02116 [cs]

[6] Jan Christian Blaise Cruz and Charibeth Cheng. 2020. Establishing Baselines for Text Classification in Low-Resource Languages. *arXiv:2005.02068 [cs]* (May 2020). arXiv:2005.02068 [cs]

[7] Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. arXiv:1912.09582 [cs]

[8] Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: A Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. ACL, 3255–3265.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs]

[10] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. arXiv:2002.06305 [cs]

[11] Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7949–7962.

[12] Andrea Esuli and Fabrizio Sebastiani. 2009. Active Learning Strategies for Multi-Label Text Classification. In *Advances in Information Retrieval*, Mohand Boughanem, Catherine Berrut, Josiane Mothe, and Chantal Soule-Dupuy (Eds.). Vol. 5478. Springer Berlin Heidelberg, 102–113.

[13] Yifan Fu, Xingquan Zhu, and Bin Li. 2013. A Survey on Instance Selection for Active Learning. *Knowledge and Information Systems* 35, 2 (May 2013), 249–283.

[14] R. A. Gilyazev and D. Yu. Turdakov. 2018. Active Learning and Crowdsourcing: A Survey of Optimization Methods for Data Labeling. *Programming and Computer Software* 44, 6 (Nov. 2018), 476–491.

[15] Antonio Gulli. 2004. AG's Corpus of News Articles. http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.

[16] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian Active Learning for Classification and Preference Learning. arXiv:1112.5745 [cs, stat]

[17] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 328–339.

[18] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing. arXiv:2108.05542 [cs]

[19] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[20] David Lowell, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical Obstacles to Deploying Active Learning. In *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. ACL, Hong Kong, China, 21–30.

[21] Katerina Margatina, Loïc Barrault, and Nikolaos Aletras. 2022. On the Importance of Effectively Adapting Pretrained Language Models for Active Learning. *arXiv:2104.08320 [cs]* (March 2022). arXiv:2104.08320 [cs]

[22] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active Learning by Acquiring Contrastive Examples. In *Proc. of the 2021 Conference on Empirical Methods in Natural Language Processing*. ACL, 650–663.

[23] Martin Mundt, Yong Won Hong, Iuliia Pliushch, and Visvanathan Ramesh. 2020. A Wholistic View of Continual Learning with Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning. arXiv:2009.01797 [cs, stat]

[24] Stephen Mussmann, Robin Jia, and Percy Liang. 2020. On the Importance of Adaptive Data Collection for Extremely Imbalanced Pairwise Tasks. arXiv:2010.05103 [cs]

[25] Anmol Nayak, Hariprasad Timmapathini, Karthikeyan Ponnalagu, and Vijendran Gopalan Venkoparao. 2020. Domain Adaptation Challenges of BERT in Tokenization and Sub-Word Representations of Out-of-Vocabulary Words. In *Proc. of the First Workshop on Insights from Negative Results in NLP*. ACL, 1–5.

[26] Harshith Padigela, Hamed Zamani, and W. Bruce Croft. 2019. Investigating the Successes and Failures of BERT for Passage Re-Ranking. arXiv:1905.01758 [cs]

[27] Sumanth Prabhu, Moosa Mohamed, and Hemant Misra. 2021. Multi-Class Text Classification Using BERT-based Active Learning. *arXiv:2104.14289 [cs]* (Sept. 2021). arXiv:2104.14289 [cs]

[28] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-Trained Models for Natural Language Processing: A Survey. *Science China Technological Sciences* 63, 10 (Oct. 2020), 1872–1897. arXiv:2003.08271

[29] Paul Röttger and Janet Pierrehumbert. 2021. Temporal Adaptation of BERT and Performance on Downstream Document Classification: Insights from Social Media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. ACL, 2400–2412.

[30] Raphael Scheible, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, and Martin Boeker. 2020. GottBERT: A Pure German Language Model. arXiv:2012.02110 [cs]

[31] Christopher Schröder and Andreas Niekler. 2020. A Survey of Active Learning for Text Classification Using Deep Neural Networks. (Aug. 2020). arXiv:2008.07267 [cs]

[32] Burr Settles. 2009. *Active Learning Literature Survey*. Technical Report. University of Wisconsin-Madison Department of Computer Sciences.

[33] Zein Shaheen, Gerhard Wohlgenannt, and Erwin Filtz. 2020. Large Scale Legal Text Classification Using Transformer Models. arXiv:2010.12871 [cs]

[34] Jingyu Shao, Qing Wang, and Fangbing Liu. 2019. Learning to Sample: An Active Learning Framework. arXiv:1909.03585 [cs, stat]

[35] Yi Song, Yuxian Gu, and Minlie Huang. 2022. Many-Class Text Classification with Matching. arXiv:2205.11409 [cs]

[36] Paul Thompson. 2001. Automatic Categorization of Case Law. In *Proc. of the 8th International Conference on Artificial Intelligence and Law (ICAIL '01)*. ACM, 70–77.

[37] Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition.

[38] Ellen M. Voorhees and Donna Harman. 2000. Overview of the Sixth Text REtrieval Conference (TREC-6). *Information Processing & Management* 36, 1 (Jan. 2000), 3–35.

[39] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. arXiv:1905.00537 [cs]

[40] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. (April 2018).

[41] Ying Wen, Weinan Zhang, Rui Luo, and Jun Wang. 2016. Learning Text Representation Using Recurrent Convolutional Neural Network with Highway Layers. (June 2016).

[42] 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. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771 [cs]

[43] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022. A Survey of Human-in-the-loop for Machine Learning. *Future Generation Computer Systems* 135 (Oct. 2022), 364–381. arXiv:2108.00941 [cs]

[44] Shanshan Yu, Jindian Su, and Da Luo. 2019. Improving BERT-Based Text Classification With Auxiliary Sentence and Domain Knowledge. *IEEE Access* 7 (2019), 176600–176612.

[45] Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-Start Active Learning through Self-supervised Language Modeling. arXiv:2010.09535 [cs]

[46] Ce Zhang, Jaeho Shin, Christopher Ré, Michael Cafarella, and Feng Niu. 2016. Extracting Databases from Dark Data with DeepDive. In *Proc. of the 2016 International Conference on Management of Data (SIGMOD '16)*. ACM, 847–859.

[47] Shujian Zhang, Chengyue Gong, Xingchao Liu, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. 2022. ALLSH: Active Learning Guided by Local Sensitivity and Hardness. arXiv:2205.04980 [cs]

[48] Anneke Zuiderwijk and Marijn Janssen. 2014. The Negative Effects of Open Government Data - Investigating the Dark Side of Open Data. In *Proc. of the 15th Annual International Conference on Digital Government Research (Dg.o '14)*. ACM, 147–152.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009