Annotation Curricula to Implicitly Train Non-Expert Annotators

Ji-Ung Lee*
UKP Lab / TU Darmstadt
lee@ukp.informatik.tu-darmstadt.de

Jan-Christoph Klie*
UKP Lab / TU Darmstadt
klie@ukp.informatik.tu-darmstadt.de

Iryna Gurevych
UKP Lab / TU Darmstadt
gurevych@ukp.informatik.tu-darmstadt.de

Annotation studies often require annotators to familiarize themselves with the task, its annotation scheme, and the data domain. This can be overwhelming in the beginning, mentally taxing, and induce errors into the resulting annotations; especially in citizen science or crowdsourcing scenarios where domain expertise is not required. To alleviate these issues, this work proposes annotation curricula, a novel approach to implicitly train annotators. The goal is to gradually introduce annotators into the task by ordering instances to be annotated according to a learning curriculum. To do so, this work formalizes annotation curricula for sentence- and paragraph-level annotation tasks, defines an ordering strategy, and identifies well-performing heuristics and interactively trained models on three existing English datasets. Finally, we provide a proof of concept for annotation curricula in a carefully designed user study with 40 voluntary participants who are asked to identify the most fitting misconception for English tweets about the Covid-19 pandemic. The results indicate that using a simple heuristic to order instances can already significantly reduce the total annotation time while preserving a high annotation quality. Annotation curricula thus can be a promising research direction to improve data collection. To facilitate future research—for instance, to adapt annotation curricula to specific tasks and expert annotation scenarios—all code and data from the user study consisting of 2,400 annotations is made available.[1]

∗ Equal contribution.

This article has been updated since publication. Please see the erratum here for details:
https://doi.org/10.1162/coli_x_00469

1 https://github.com/UKPLab/annotation-curriculum.

Submission received: 7 June 2021; revised version received: 9 December 2021; accepted for publication: 21 December 2021.

https://doi.org/10.1162/coli_a_00436

1. Introduction

Supervised learning and, consequently, annotated corpora are crucial for many downstream tasks to train and develop well-performing models. Despite improvements of models trained in a semi- or unsupervised fashion (Peters et al. 2018; Devlin et al. 2019), they still substantially benefit from labeled data (Peters, Ruder, and Smith 2019; Gururangan et al. 2020). However, labels are costly to obtain and require domain experts or a large crowd of non-expert annotators (Snow et al. 2008).

Past research has mainly investigated two approaches to reduce annotation cost and effort (often approximated by annotation time); namely, **active learning** and **label suggestions**. Active learning assumes that resources for annotating data are limited and aims to reduce the number of labeled instances by only annotating those that contribute most to model training (Lewis and Gale 1994; Settles 2012). This often results in sampled instances that are more difficult to annotate, putting an increased cognitive load on annotators, and potentially leading to a lower agreement or an increased annotation time (Settles, Craven, and Friedland 2008). Label suggestions directly target annotators by providing them with suggestions from a pre-trained model. Although they are capable of effectively reducing the annotation time (Schulz et al. 2019; Klie, Eckart de Castilho, and Gurevych 2020; Beck et al. 2021, they bear the risk of biasing annotators toward the (possibly erroneous) suggested label (Fort and Sagot 2010). Both these shortcomings render existing approaches better suited for domain-expert annotators who are less burdened by difficult annotation instances and are less prone to receiving erroneous label suggestions than non-expert annotators. Overall, we can identify a lack of approaches that (1) are less distracting or biased than label suggestions and (2) can also ease the annotation process for non-expert annotators. Especially, the increasing popularity of large-scale, crowdsourced datasets (Bowman et al. 2015; Sakaguchi et al. 2021) further amplifies the need for training methods that can also be applied in non-expert annotator scenarios (Geva, Goldberg, and Berant 2019; Nie et al. 2020; Rogers 2021).

One key element that has so far not been investigated in annotation studies is the use of a curriculum to *implicitly* teach the task to annotators during annotation. The **learning curriculum** is a fundamental concept in educational research that proposes to order exercises to match a learner's proficiency (Vygotsky 1978; Krashen 1982) and has even motivated training strategies for machine learning models (Bengio et al. 2009). Moreover, Kelly (2009) showed that such learning curricula can also be used to teach learners implicitly. Similarly, the goal of **annotation curricula** (AC) is to provide an ordering of instances during annotation that is optimized for learning the task. We conjecture that a good annotation curriculum can implicitly teach the task to annotators—for instance, by showing easier annotation instances before more difficult ones—consequently reducing the cognitive strain and improving annotation speed and quality. In contrast to active learning, which may result in only sampling instances that are difficult to annotate, they explicitly emphasize the needs of a human annotator and gradually familiarize them with the annotation task. Compared to label suggestions, they are less distracting as they do not bear the risk of providing erroneous suggestions from imperfect models, making them well-suited for non-expert annotation scenarios. Furthermore, AC do not require study conductors to adapt existing annotator training processes or annotation guidelines and hence, can complement their annotation project. To provide a first assessment for the

viability of such annotation curricula, we investigate the following three research questions:

RQ1. Does the order in which instances are annotated impact the annotations in terms of annotation time and quality?

RQ2. Do traditional heuristics and recent methods for assessing the reading difficulty already suffice to generate curricula that improve annotation time or quality?

RQ3. Can the generation of annotation curricula be further alleviated by interactively trained models?

We first identify and formalize two essential parts to deploy AC: (1) a "strategy" that defines how instances should be ordered (e.g., by annotation difficulty) and (2) an "estimator" that ranks them accordingly. We instantiate AC with an "easy-instances-first" strategy and evaluate heuristic and interactively trained estimators on three English datasets that provide annotation time which we use as an approximation of the annotation difficulty for evaluation. Finally, we apply our strategy and its best estimators in a carefully designed user study with 40 participants for annotating English tweets about the Covid-19 pandemic. The study results show that the ordering in which instances are annotated can have a statistically significant impact on the outcome. We furthermore find that annotators who receive the same instances in an optimized order require significantly less annotation time while retaining a high annotation quality. Our contributions are:

C1. A novel approach for training non-expert annotators that is easy to implement and is complementary to existing annotator training approaches.

C2. A formalization of AC for sentence- and paragraph-labeling tasks with a strategy that orders instances from easy to difficult, and an evaluation for three heuristics and three interactively trained estimators.

C3. A first evaluation of AC in a carefully designed user study that controls for external influences including:

a) An implementation of our evaluated annotation curriculum strategies and 2,400 annotations collected during our human evaluation study.

b) A production-ready implementation of interactive AC in the annotation framework INCEpTION (Klie et al. 2018) that can be readily deployed.

Our evaluation of different heuristics and interactively trained models further reveals additional factors—such as the data domain and the annotation task—that can influence their aptitude for AC. We thus appeal to study conductors to publish the annotation order and annotation times along with their data to allow future studies to better investigate and develop task- and domain-specific AC.

2. Related Work

Most existing approaches that help with data collection focus on either active learning or label suggestions. Other researchers also investigate tackling annotation task within the context of gamification and introduce different levels of difficulty.

Active Learning. Active learning has widely been researched in terms of model-oriented approaches (Lewis and Gale 1994), Roy and McCallum 2001; Gal, Islam, and Ghahramani 2017; Siddhant and Lipton 2018; Kirsch, van Amersfoort, and Gal 2019), data-oriented approaches (Nguyen and Smeulders 2004; Zhu et al. 2008; Huang, Jin, and Zhou 2010; Wang et al. 2017), or combinations of both (Ash et al. 2020; Yuan, Lin, and Boyd-Graber 2020). Although several works investigate annotator proficiency—which is especially important for crowdsourcing—their main concern is to identify noisy labels or erroneous annotators (Laws, Scheible, and Schütze 2011; Fang et al. 2012; Zhang and Chaudhuri 2015) or distribute tasks between workers of different proficiency (Fang, Yin, and Tao 2014; Yang et al. 2019). Despite the large amount of research in active learning, only a few studies have considered annotation time as an additional cost variable in active learning (Settles, Craven, and Friedland 2008) and even found that active learning can negatively impact annotation time (Martínez Alonso et al. 2015). Other practical difficulties for deploying active learning in real annotation studies stem from additional hyperparameters that are introduced, but seldom investigated (Lowell, Lipton, and Wallace 2019). In contrast, AC also work well with simple heuristics, allowing researchers to pre-compute the order of annotated instances.

Label Suggestions. Label suggestions have been considered for various annotation tasks in NLP, such as in part-of-speech tagging for low-resource languages (Yimam et al. 2014), interactive entity-linking (Klie, Eckart de Castilho, and Gurevych 2020), or identifying evidence in diagnostic reasoning (Schulz et al. 2019). Especially for tasks that require domain-specific knowledge such as in the medical domain, label suggestions can substantially reduce the burden on the annotator (Lingren et al. 2014). However, they also inherently pose the risk of amplifying annotation biases due to the anchoring effect (Turner and Schley 2016). Whereas domain experts may be able to reliably identify wrong suggestions and provide appropriate corrections (Fort and Sagot 2010), this cannot be assumed for non-experts. This renders label suggestions a less viable solution to ease annotations in non-expert studies where incorrect label suggestions may even distract annotators from the task. In contrast, changing the ordering in which instances are annotated by using AC is not distracting at all.

Annotation Difficulty. Although difficulty estimation is crucial in human language learning, for instance, in essay scoring (Mayfield and Black 2020) or text completion exercises (Beinborn, Zesch, and Gurevych 2014; Loukina et al. 2016; Lee, Schwan, and Meyer 2019), it is difficult to achieve in annotation scenarios due to the lack of ground truth, commonly resulting in a post-annotation analysis for model training (Beigman Klebanov and Beigman 2014; Paun et al. 2018). To consider the difficulty of annotated instances, a concept that has recently been explored for (annotation) games with a purpose, is **progression**. It allows annotators to progress through the annotation study similar to a game—by acquiring specific skills that are required to progress to the next level (Sweetser and Wyeth 2005). Although several works have shown the efficiency of progression in games with a purpose (Madge et al. 2019; Kicikoglu et al. 2020) and even in crowdsourcing (Tauchmann, Daxenberger, and Mieskes 2020), this does not

necessarily benefit individual workers, as less-skilled workers are either filtered out or asked to "train" on additional instances. Moreover, implementing progression poses a substantial burden on researchers due to the inclusion of game-like elements (e.g., skills and levels), or at minimum, the separation of the data according to difficulty and, furthermore, a repeated evaluation and reassignment of workers. In contrast, reordering instances of a single set according to a given curriculum can already be achieved with low effort and can even be implemented complementary to progression.

3. Annotation Curriculum

We first specify the type of annotation tasks investigated in this work, and then formalize AC with the essential components that are required for generating appropriate annotation curricula. Finally, we instantiate an easy-instances-first strategy and define the estimators that we use to generate a respective curriculum.

3.1 Annotation Task

In this work, we focus on sentence- and paragraph-level annotation tasks that do not require any deep domain-expertise and hence are often conducted with non-expert annotators.[2] Such annotation tasks often use a simple annotation scheme limited to a small set of labels, and have been used to create datasets across various research areas, for instance, in sentiment analysis (Pak and Paroubek 2010), natural language inference (Bowman et al. 2015), and argument mining (Stab et al. 2018).

Task Formalization. We define an annotation task as being composed of a set of unlabeled instances $x \in \mathcal{U}$ that are to be annotated with their respective labels $y \in \mathcal{Y}$. We focus on instances x that are either a sentence or a paragraph and fully annotated by an annotator a. Note that for sequence labeling tasks such as named entity recognition, y is not a single label but a vector composed of the respective token-level labels. However, in such tasks, annotations are still often collected for a complete sentence or paragraph at once to provide annotators with the necessary context (Tomanek and Hahn 2009).

3.2 Approach

Figure 1 provides a general overview of AC. Given a set of unlabeled instances $x \in \mathcal{U}$, we define a strategy $\mathcal{S}$ that determines the ordering in which annotated instances should be presented (easy-instances-first). We then specify "adaptive" and "non-adaptive" estimators $f(\cdot)$ that approximate the true annotation difficulty. In this work, we focus on task-agnostic estimators that can easily be applied across a wide range of tasks and leave the investigation on task-specific estimators—which may have higher performance but also require more implementation effort from study conductors—for future work.[3] Depending on the estimator, we then order the annotated instances either beforehand (non-adaptive), or select them iteratively at each step based on the predictions of an interactively trained model (adaptive).

2 We discuss AC strategies that may be better suited for domain experts in Section 6.
3 We discuss some ideas for task-specific estimators in Section 6.

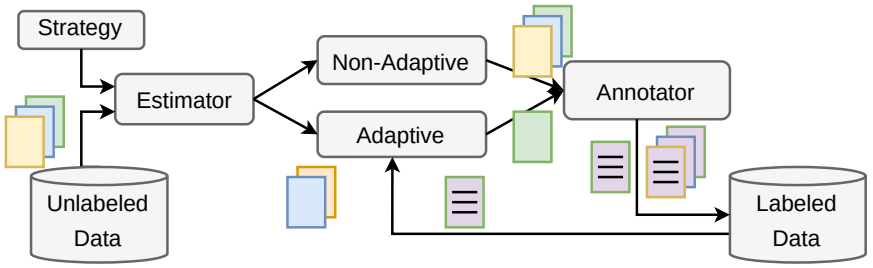


Figure 1
Annotation curricula. First, we define a strategy for ordering instances by annotation difficulty (i.e., easy-first). We then implement estimators that perform the ordering. Estimators can either be non-adaptive (e.g., heuristics) or adaptive (trained models). Finally, annotators receive instances according to the resulting curriculum.

Formalization. Ideally, an annotation curriculum that optimally introduces annotators to the task would minimize (1) annotation effort and (2) error rate (i.e., maximize annotation quality). As the annotation error can only be obtained post-study, we can only use annotation effort, approximated by annotation time, for our formalization; however, we conjecture that minimizing annotation time may also have a positive impact on annotation quality (given that the annotators remain motivated throughout their work). To further reduce noise factors during evaluation, we focus on annotation studies that involve a limited number of instances (in contrast to active learning scenarios that assume an abundance of unlabeled data). We thus formalize annotation curriculum as the task of finding the optimal curriculum $\mathcal{C}^*$ out of all possible curricula $\mathcal{C}$ (i.e., permutations of $\mathcal{U}$) for a finite set of unlabeled instances $\mathcal{U}$ that minimizes the total annotation time $\mathcal{T}$; namely, the sum of individual annotation times $t_i \in \mathbb{R}^+$ for all instances $x_i \in \mathcal{U}$ with i denoting the i-th annotated instance:

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \sum_{i=1}^{|\mathcal{U}|} a_i(x_i|x_0 \dots x_{i-1}) \tag{1}$$

where $a_i : \mathcal{U} \rightarrow \mathcal{T}$ describes the annotator after annotating $i-1$ instances.

Strategy. Due to the large number of $n!$ possible curricula $\mathcal{C}$ resulting from $n = |\mathcal{U}|$ instances, solving Equation 1 is intractable for large n even if $a(\cdot)$ was known. We can furthermore only assess the true effectiveness of a curriculum $\mathcal{C}$ post-study, making it impossible to find the optimal curriculum $\mathcal{C}^*$ beforehand. We hence require a strategy $\mathcal{S} \sim \mathcal{C}^*$ that specifies how instances of $\mathcal{U}$ should be ordered optimally. Similar to educational approaches, we rely on estimating the "difficulty" of an instance to generate our curriculum (Taylor 1953; Beinborn, Zesch, and Gurevych 2014; Lee, Schwan, and Meyer 2019). In this work, we investigate an easy-instances-first strategy that has been shown to be a reasonable strategy in previous work (Tauchmann, Daxenberger, and Mieskes 2020); thereby sorting instances in ascending order according to their difficulty. Our $\mathcal{C}^*$ is thus approximated by the ordered set $\mathcal{S} = \{x_1, \dots, x_n | \forall x_{1 \leq i \leq n} \in \mathcal{S} : f(x_i) \leq f(x_{i+1})\}$ with $f(\cdot)$ being the difficulty estimator.

Non-adaptive Estimators. We define non-adaptive estimators as heuristics or pre-trained models that are not updated interactively. The respective annotation curriculum can thus be pre-computed and does not impose any additional changes to the underlying annotation platform. To estimate the annotation difficulty, non-adaptive estimators define a scoring function $f_{\bar{a}} : \mathcal{U} \rightarrow \mathbb{R}$. In this work, we evaluate non-adaptive estimators that are commonly used in readability assessment to score the reading difficulty of a text (Xia, Kochmar, and Briscoe 2016; Deutsch, Jasbi, and Shieber 2020). Although they are not capable of capturing any task-specific difficulties, they have the advantage of being applicable to a wide range of tasks with low effort for study conductors. The following heuristics and pre-trained models are investigated to obtain difficulty estimations for the easy-instances-first curriculum:

Sentence Length (sen**)** The number of tokens in a sentence averaged across the whole document (i.e., the number of tokens for single sentence instances).

Flesch-Kincaid (FK**)** A readability score based on the number of words, syllables, and sentences (Kincaid et al. 1975).

Masked Language Modeling Loss (mlm**)** As shown in recent work, the losses of a masked language model may be used to obtain an assessment of text complexity (Felice and Buttery 2019). We use the implementation of Salazar et al. (2020).

Adaptive Estimators. While simple heuristics or annotator-unaware models allow us to pre-compute annotation curricula, they do not consider any user-specific aspect that may influence the difficulty estimation (Lee, Meyer, and Gurevych 2020). Consequently, the resulting curriculum may not provide the optimal ordering for a specific annotator. To select the instance with the most appropriate difficulty for an annotator $a_i(\cdot)$ at the i-th iteration, we use a model $\theta_i(\cdot) \sim a_i(\cdot)$ that is updated with an increasing number of annotated instances. We conjecture that using $\theta(\cdot)$ to predict the relative difficulty—in contrast to non-adaptive estimators that provide an absolute difficulty estimation—may be more robust to task-specific influences as they are inherited in all instances annotated by $a(\cdot)$. When training adaptive estimators, we use annotation time to approximate the difficulty of a specific instance due to its availability in any annotation scenario. At iteration i, we thus train the model $\theta_i : \mathcal{L} \rightarrow \mathcal{T} \subseteq \mathbb{R}^+$ to predict the annotation times $t \in \mathcal{T}$ for all labeled instances $\hat{x} \in \mathcal{L}$. Similar to active learning, we now encounter a decreasing number of unlabeled instances and an increasing number of labeled instances. The resulting model is then used to estimate the annotation time for all unlabeled instances $x \in \mathcal{U}$. The resulting scoring function is now defined as $f_a : \theta_i, \mathcal{U} \rightarrow \mathbb{R}^+$. Finally, we select instance $x^* \in \mathcal{U}$ with the minimal rank according to f_a.

$$x^* = \arg\min_{f_a} \theta_i(x) \tag{2}$$

Following our strategy $\mathcal{S}$, this results in selecting instances for annotation that have the lowest predicted annotation time. We specifically focus on regression models that can be trained efficiently in-between annotation and work robustly in low-data scenarios. We choose Ridge Regression, Gaussian Process Regression, and GBM Regression.

4. Evaluation with Existing Datasets

To identify well-performing non-adaptive and adaptive estimators, we first evaluate AC on existing datasets in an offline setting. We focus on datasets that provide annotation time which is used to approximate the annotation difficulty during evaluation (to address the lack of gold labels in actual annotation scenarios). Following Settles, Craven, and Friedland (2008), we conjecture that instances with a higher difficulty require more time to annotate. For comparison, we then compute the correlations between different orderings generated according to our easy-instances-first strategy using text difficulty heuristics (non-adaptive) and interactively trained models (adaptive) with the annotation time (approximated annotation difficulty). We evaluate our estimators in two setups:

Full We evaluate how well adaptive and non-adaptive estimators trained on the whole training set correlate with the annotation time of the respective test set (upper bound).

Adaptive We evaluate the performance of adaptive estimators in an interactive learning scenario with simulated annotators and an increasing number of training instances.

4.1 Datasets

Overall, we identify three NLP datasets that provide accurate annotation time for individual instances along with their labels:

Muc7$_T$ Tomanek and Hahn (2009) extended the MUC7 corpus that consists of annotated named entities in English Newswire articles. They reannotated the data with two annotators A and B while measuring their annotation time per sentence.

SigIE is a collection of email signatures that was tagged by Settles, Craven, and Friedland (2008) with twelve named entity types typical for email signatures such as phone number, name, and job title.

SPEC The same authors (Settles, Craven, and Friedland 2008) further annotated sentences from 100 English PubMed abstracts according to their used language (speculative or definite) with three annotators.

Table 1
Annotation task (ST for sequence tagging, Cl for classification) and the number of instances per dataset and split. $\mu_{|\mathcal{D}|}$ denotes the average instance length in characters and μ_t the average annotation time. $\sigma_{|\mathcal{D}|}$ and σ_t denotes the standard deviation, respectively. Across all datasets, annotation time is reported for annotating the whole instance (i.e., not for individual entities).

Name	Task	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{D}_{\textbf{train}}\rvert$	$\lvert\mathcal{D}_{\textbf{dev}}\rvert$	$\lvert\mathcal{D}_{\textbf{test}}\rvert$	$\mu_{\lvert\mathcal{D}\rvert}$	$\sigma_{\lvert\mathcal{D}\rvert}$	μ_t	σ_t
Muc7$_T$ A	ST	3,113	2,179	467	467	133.7	70.8	5.4	3.9
Muc7$_T$ B	ST	3,113	2,179	467	467	133.7	70.8	5.2	4.2
SigIE	ST	251	200	–	51	226.4	114.8	27.0	14.7
SPEC	Cl	850	680	–	170	160.4	64.2	22.7	12.4

Table 1 provides an overview of the used datasets. It can be seen that Muc7_T is the largest corpus ($|\mathcal{D}|$); however, it is also the one that consists of the shortest instances on average ($\mu_{|\mathcal{D}|}$). Furthermore, Muc7_T also has the lowest annotation times (μ_t) and a low standard deviation (σ_t). Comparing the number of entities per instance between Muc7_T (news articles) and SigIE (email signatures) shows their differences with respect to their domains with an average number of 1.3 entities ($\sigma = 1.4$) in Muc7_T and 5.3 entities ($\sigma = 3.0$) in SigIE. Moreover, we find that the SigIE corpus has a higher ratio of entity tokens (40.5%) than Muc7_T (8.4%), which may explain the long annotation time. Interestingly, the binary sentence classification task SPEC ("speculative" or "definite") also displays a substantially longer annotation time compared to Muc7_T (on average, more than four times), which may also indicate a higher task difficulty or less proficiency of the involved annotators.

Data splits. For Muc7_T, we focus on the annotations of the first annotator Muc7_T A; using Muc7_T B yields similar results. For SPEC, we use ALL.DAT for our experiments. None of the aforementioned datasets provide default splits. We hence create 80-20 train-test splits of SPEC and SigIE for our experiments. To identify the best hyperparameters of our adaptive estimators, we split the largest corpus (Muc7_T) into 70-15-15 train-dev-test splits. All splits are published along with the code and data.

4.2 Experimental Setup

Our goal is to evaluate how well the ordering generated by an estimator correlates with the annotation time provided in the respective datasets.

Evaluation Metrics. We evaluate all estimators by measuring Spearman's ρ between the true and generated orderings of all instances in the test data. We obtain the generated ordering by sorting instances according to the predicted annotation time. For our adaptive estimators that explicitly learn to predict the annotation time, we further report the mean absolute error (MAE), the rooted mean squared error (RMSE), and the coefficient of determination (R^2).

Models and Features. For an effective deployment in interactive annotation scenarios, we require models that are capable of fast training and inference. We additionally consider the amount of computational resources that a model requires as these pose further limitations for the underlying annotation platform. Consequently, fine-tuning large language models such as BERT is infeasible as they require long training times and a large amount of computational resources.[4] Instead, we utilize a combination of neural embeddings obtained from a large pre-trained language model combined with an efficient statistical model. As our goal is to predict the total time an annotator requires to annotate an instance (i.e., a sentence or a paragraph), we further require a means to aggregate token- or subtoken-level embeddings that are used in recent language models (Sennrich, Haddow, and Birch 2016). One such solution is S-BERT (Reimers and Gurevych 2019), which has shown high performance across various tasks. Moreover, Reimers and Gurevych (2019) provide S-BERT for a variety of BERT-based models, allowing future study conductors to easily extend our setup to other languages

4 Note that using such models would require an annotation platform to either deploy its own GPU or buy additional computational resources from external providers.

Table 2
Hyperparameter tuning for adaptive estimators. We train on Muc7_T A and evaluate on its development set. t denotes the total time for training and prediction on the whole dataset. Best parameters are marked by * and the best scores are highlighted in **bold**. We report the mean absolute error (MAE), the rooted mean squared error (RMSE), Spearman's ρ, and the coefficient of determination (R^2).

Name	Features	MAE	RMSE	R^2	ρ	t
RR($\alpha = 0.5$)	BOW	1.85	2.96	0.47	0.73	0.42
RR($\alpha = 0.5$)	S-BERT	1.92	2.84	0.51	0.79	**0.04**
RR($\alpha = 1$)	BOW	1.80	2.91	0.49	0.74	0.41
RR($\alpha = 1$) *	S-BERT	1.89	2.82	0.52	0.79	0.04
GP(kernel=Dot + White)	BOW	1.82	2.93	0.48	0.74	257.67
GP(kernel=Dot + White) *	S-BERT	**1.80**	**2.76**	**0.54**	**0.81**	14.35
GP(kernel=RBF(1.0)	BOW	5.33	6.71	−1.73	−0.12	300.38
GP(kernel=RBF(1.0)	S-BERT	5.33	6.71	−1.73	−0.12	32.66
GBM	BOW	2.07	3.26	0.36	0.68	0.25
GBM *	S-BERT	1.83	2.83	0.52	0.79	2.98

and specific tasks. For computational efficiency, we use the *paraphrase-distilroberta-base-v1* model, which utilizes a smaller, distilled RoBERTa model (Sanh et al. 2019). As a comparison to S-BERT, we further evaluate bag-of-words (BOW) features for all three models (cf. Table 2). For the Ridge Regression (RR), Gaussian Process Regression (GP), and GBM Regression (GBM) models, we use the implementations of Pedregosa et al. (2011) and Ke et al. (2017).

Hyperparameter Tuning. We use the full experimental setup to identify the best performing parameters for our experiments using simulated annotators. We evaluate different values for regularization strength (α) for RR and we evaluate different kernel functions for GP. To ensure that the required training of our adaptive estimators does not negatively affect the annotations due to increased loading times and can be realistically performed during annotation, we further measure the overall training time (in seconds). We use the development split of Muc7_T A to tune our hyperparameters for all models used across all datasets. Considering the small number of training instances in both datasets, we do not tune SigIE- or SPEC-specific hyperparameters. All experiments were conducted using an *AMD Ryzen 5 3600*. Table 2 shows the results of our hyperparameter tuning experiments. Overall, we find that S-BERT consistently outperforms BOW in terms of Spearman's ρ. As the result of the hyperparameter tuning, we use S-BERT embeddings as input features and evaluate GP with a combined dot- and white-noise kernel and RR with $\alpha = 1$ in our adaptive experiments.

4.3 Experimental Results

We first report our experimental results for the full and adaptive setup. For conducting our experiments with simulated annotators, we use the best performing models from our hyperparameter tuning of the respective models on the Muc7_T dataset and report the results of the best performing models.

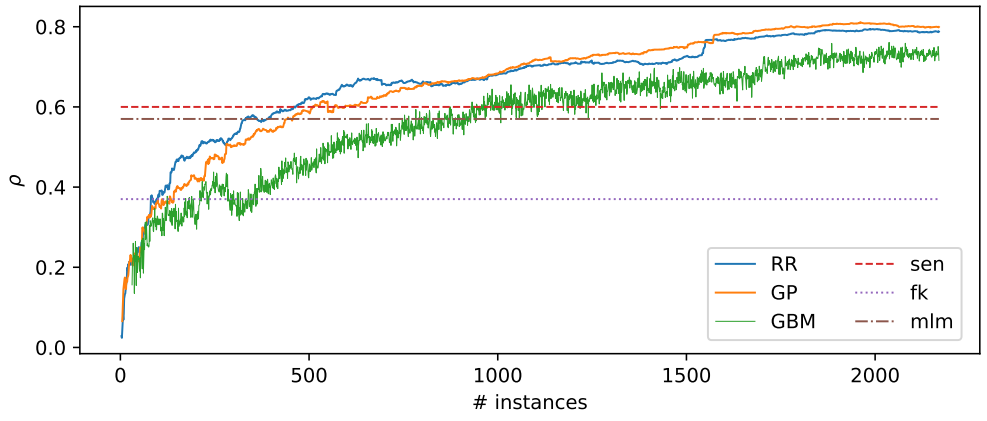
Table 3
Performance of the best performing adaptive estimators on the four datasets ($Muc7_T$ provides annotation times from two different annotators A and B) trained on the respective train and evaluated on their test splits. We report the mean absolute error (MAE), the rooted mean squared error (RMSE), the coefficient of determination (R^2), and Spearman's ρ.

Name	Model	MAE	RMSE	R^2	ρ	t
$Muc7_T$ A	RR	1.87	2.68	0.56	0.80	0.15
	GP	1.79	2.66	0.57	0.82	7.23
	GBM	1.95	2.97	0.47	0.75	3.40
$Muc7_T$ B	RR	2.19	3.42	0.44	0.79	0.02
	GP	2.08	3.37	0.46	0.81	8.85
	LGBM	2.13	3.50	0.41	0.75	2.90
SigIE	RR	7.96	9.50	0.46	0.73	0.00
	GP	7.62	9.60	0.44	0.70	0.08
	GBM	8.22	10.84	0.29	0.55	0.14
SPEC	RR	9.63	13.86	−0.14	0.50	0.03
	GP	7.63	12.07	0.14	0.51	0.73
	GBM	8.05	12.50	0.07	0.35	1.70

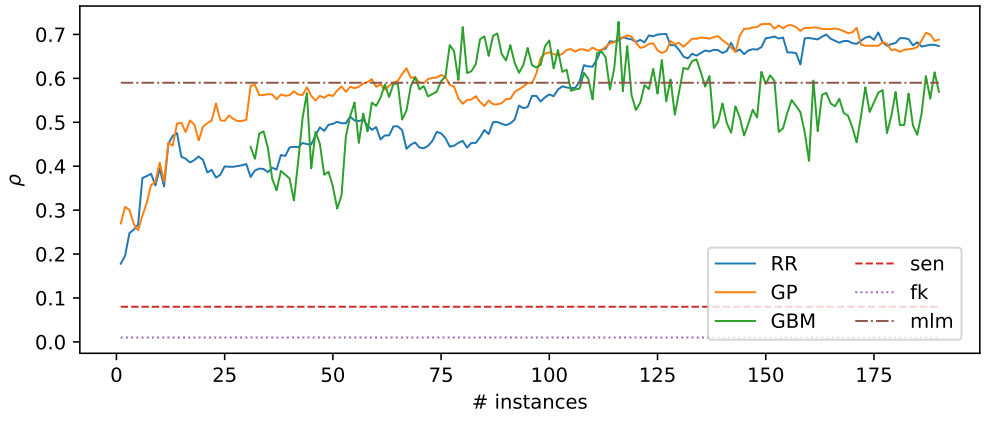
Full Results. Table 4 shows the results for the heuristic estimators and regression models evaluated on the test split of each dataset. We find that heuristics that mainly consider length-based features (sen and FK) are not suited for the SigIE data that consist of email signatures. One reason for this may be the different text type of email signatures in comparison to Newswire articles and PubMed abstracts. More specifically, analyzing the ratio between non-alphabetical or numeric characters (excluding @ and .) and other characters shows that SigIE contains a substantial number of characters that are used for visually enhancing the signature (some are even used in text art). Overall, 29.9% of the characters in SigIE are non-alphabetical or numeric, in contrast to 16.7% in SPEC and 19.9% in $Muc7_T$.[5] Considering that only 1.7% of them appear within named entities in SigIE (such as + in phone numbers) most of them rather introduce noise especially for length-based features such as sen and FK. On $Muc7_T$ and SPEC, all three heuristics produce an ordering that correlates with annotation time to some extent. On average, mlm is the best performing and most robust heuristic across all three datasets. For our adaptive estimators, RR and GP both similarly outperform GBM in terms of Spearman's ρ. However, we can find that GP consistently outperforms RR and GBM in terms of MAE and RMSE, as well as in terms of R^2 on $Muc7_T$ and SPEC. We report the extensive results in Table 3.

Adaptive Results. To evaluate the performance of adaptive estimators with increasing numbers of annotated instances, we perform experiments with simulated annotators. At each iteration, we use a model trained on the already-annotated data to select the instance with the lowest predicted annotation time (randomly in the first iteration). The simulated annotator then provides the respective gold annotation time, which is then added to the training set. Finally, the model is re-trained and evaluated on the test data. These steps are repeated until all instances are annotated. Figure 2 shows the Spearman's ρ performance of all three models after each iteration across all datasets. We

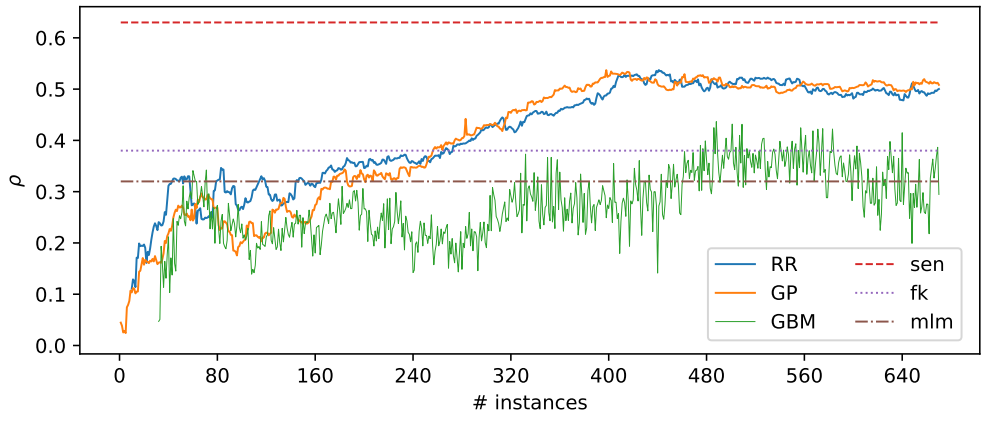
5 The Twitter data we introduce in Section 5 consist of 20.7% non-alphabetical or numeric characters.



(a) Muc7$_T$ A



(b) SigIE



(c) SPEC

Figure 2
Experimental results of our adaptive estimators with simulated annotators. Horizontal lines show the performance of the respective non-adaptive estimators.

can observe that all models display a rather steep learning curve after training on only a few examples, despite suffering from a cold start in early iterations. Moreover, we find that GP and RR are capable of outperforming mlm consistently after 100–500 instances. GBM shows the weakest performance and is consistently outperformed by the other models for Muc7_T and SPEC while being rather noisy. Although we find that non-adaptive estimators can suffice, especially in early iterations, our experiments also show the potential of adaptive estimators with an increasing number of annotations. This indicates that hybrid approaches that combine non-adaptive and adaptive estimators could be an interesting direction for future work. For instance, one may consider using non-adaptive estimators in early stages until a sufficient number of annotated instances are available to train more reliable adaptive estimators. Another approach could be to combine the rankings of different estimators, for instance, via Borda count (Szpiro 2010) or learn a weighting of the individual estimators.

5. Human Evaluation

To evaluate the effectiveness of our easy-instances-first AC with real annotators, we conduct a user study on a classification task for English tweets and analyze the resulting annotations in terms of *annotation time* and *annotation quality*. We design the study to not require domain-expertise and conduct it with citizen science volunteers.[6]

Hypothesis. We investigate the following hypothesis: Annotators who are presented with easy instances first and then with instances that gradually increase in terms of annotation difficulty require less annotation time or have improved annotation quality compared with annotators who receive the same instances in a random order.

5.1 Study Design

A careful task and data selection are essential to evaluate AC, as our goal is to measure differences that solely result from a different ordering of annotated instances. We also require instances with varying difficulty, further restricting our study design in terms of task and data.

Data Source. To avoid compromising the study results due to noisy data, we use an existing corpus that has been carefully curated and provides gold labels for evaluating the annotation quality. To involve non-expert annotators, we further require data that do not target narrow domains or require expert knowledge. As such, tasks such as identifying part-of-speech tags would substantially reduce the number of possible study participants due to the required linguistic knowledge. We identify COVIDLies (Hossain et al. 2020) as a suitable corpus due to the current relevance and the high media-coverage of the Covid-19 pandemic; ensuring a sufficient number of participants who are well-versed with the topic. The corpus consists of English tweets that have been annotated by medical experts with one out of 86 common misconceptions about the Covid-19 pandemic. Each instance consists of a tweet-misconception pair and if the tweet "agrees," "disagrees," or has "no stance" toward the presented misconception.

6 We provide a statement regarding the conduct of ethical research after the conclusion.

Annotation Task. Using the COVIDLies corpus as our basis, we define a similar task that is better suited for lay people and that allows us to explicitly control the annotation difficulty. We restrict the task to identifying the most appropriate misconception out of six possible choices. Furthermore, we only include tweets that agree with a misconception (i.e., we do not ask for a stance annotation) to avoid interdependencies between stance and misconception annotations that may introduce additional noise to the results and put an unnecessary burden on the participants.[7] To exclude further sources of noise for our study, we manually check all tweets and remove all duplicates (possibly due to retweets) and hyperlinks to increase readability and avoid distractions. We also remove all tweets that were malformed (i.e., ungrammatical or containing several line breaks) or linked to misconceptions with less than five semantically similar candidates that could serve as distractors.[8] For the final selection, we choose the 60 shortest tweets.

Distractor Selection. The goal of the study is to observe effects that solely result from the ordering of instances with varying annotation difficulty. Hence, we need to ensure that annotated instances correspond to specific difficulties and are balanced equally for each participant. To control the annotation difficulty, we construct five possible sets of misconceptions for each instance that are presented to the annotator; each corresponding to a respective difficulty-level ranging from "very easy" to "very difficult." Each set consists of the expert-selected misconception and five additional misconceptions that serve as distractors which are commonly used in cloze-tests (Taylor 1953). Following existing research on automated cloze-test generation, we focus on **semantic similarity** to generate distractor subsets (Agarwal and Mannem 2011; Mostow and Jang 2012; Yeung, Lee, and Tsou 2019) and manually create one set of five semantically dissimilar and one set of five semantically similar misconceptions for each misconception.[9] As semantically dissimilar distractors are much easier to identify than semantically similar ones (Mostow and Jang 2012), we can manipulate annotation difficulty by adapting the number of semantically similar distractors; that is, starting from the set of dissimilar (very easy) misconceptions, we can gradually increase the difficulty by replacing a dissimilar misconception with a similar one until only the set of similar (very difficult) misconceptions remains. Figure 3 shows a tweet from our user study with its respective easy and difficult misconception sets. As can be seen, the difficult misconception set consists of two more semantically similar misconceptions. Especially notable is the third misconception, which states the opposite of the tweet's misconception but with a similar wording.

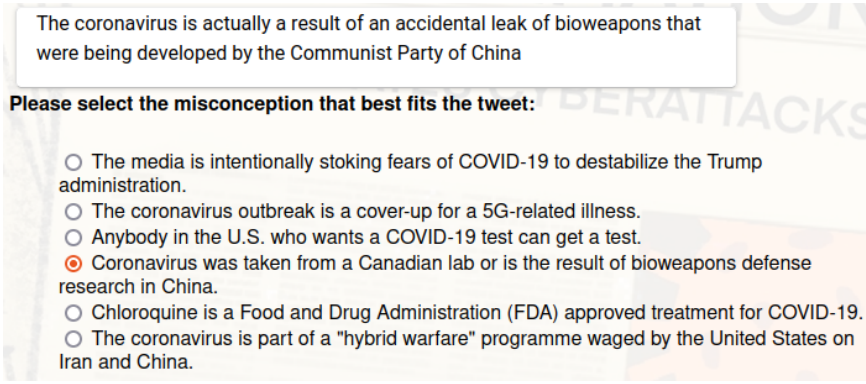
5.2 Study Setup

We set up our evaluation study as a self-hosted Web application that is only accessible during the study (one week). Participants can anonymously participate with a self-chosen, unique study key that allows them to request the deletion of their provided data at a later point. Upon registration, they are informed about the data presented and

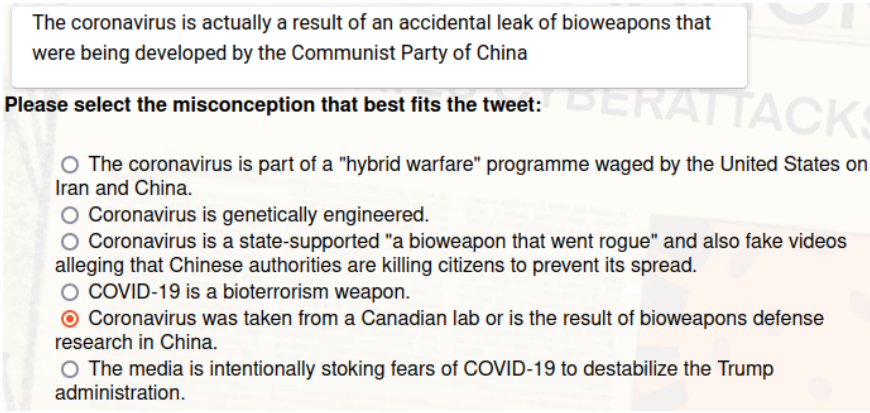
7 We experimented with including stance annotations (positive, negative, or neutral) during early stages of our study setup but removed them due to a substantially increased overall annotation difficulty.

8 The sets of similar misconceptions were manually created as explained in the next paragraph.

9 Initially, we also investigated the use of recent automated approaches to create those subsets (Gao, Gimpel, and Jensson 2020). However, the resulting subsets rather targeted syntactic instead of semantic similarity. One reason for this may be that approaches to generate cloze-tests consider only single-token gaps whereas the misconceptions consist of several words that form a descriptive statement.



(a) Easy Example



(b) Difficult Example

Figure 3
Example tweet from the user study with an easy misconception set (used in the study) and a difficult misconception set.

collected in the study, its further use, and the purpose of the study. Before collecting any data, participants are explicitly asked for their informed consent. Overall, we recruited 40 volunteers who provided their informed consent to participate in our study and annotated 60 instances each.

Participants. Our volunteers come from a variety of university majors, native languages, English proficiency, and annotation experience backgrounds. All participants provided a rather high self-assessment of English proficiency, with the lowest proficiency being intermediate (B1) provided by only one participant. Seventy percent of the participants stated an English proficiency-level of advanced (C1) or proficient (C2). Most participants have a higher level of education and are university graduates with either a Bachelor's or Master's degree; however, none of them have a medical background, which may have given them an advantage during the annotation study. Upon completing the annotations, all participants received a questionnaire including general questions about their previous annotation experience and perceived difficulty of the task (cf. Section 5.5).

Table 4
Spearman's ρ between test data and the orderings generated by the evaluated heuristics and adaptive models.

Dataset	sen	FK	mlm	RR	GP	GBM
$Muc7_T$ A	0.60	0.37	0.57	0.80	**0.82**	0.75
$Muc7_T$ B	0.60	0.38	0.55	0.79	**0.81**	0.75
SigIE	0.08	0.01	0.59	**0.73**	0.70	0.55
SPEC	**0.63**	0.38	0.32	0.50	0.51	0.35
Average	0.48	0.29	0.52	0.71	0.71	0.60

Ordering Strategy. All participants are randomly assigned to one out of four groups (ten participants per group), each corresponding to a strategy that leads to a different ordering of annotated instances. We investigate the following strategies:

Random is the control group that consists of randomly ordered instances.

AC_{mlm} uses the masked language modeling loss. It is a pre-computed, heuristic estimator and had (on average) the highest and most stable correlation to annotation time in our experiments with simulated annotators.

AC_{GP} uses a Gaussian Process that showed the highest performance on the sentence-labeling task (SPEC) in our simulated annotator experiments (cf. Table 4). It is trained interactively to predict the annotation time. We train a personalized model for each annotator using S-BERT embeddings of the presented tweet.

AC_{gold} consists of instances explicitly ordered from very easy to very difficult using the pre-defined distractor sets. Although such annotation difficulties are unavailable in real-world annotation studies, it provides an upper-bound for the study.

Control Instances. To provide a fair comparison between different groups, we further require participants to annotate instances that quantify the difference with respect to prior knowledge and annotation proficiency. For this, we select the first ten instances and present them in the same order for all annotators. To avoid interdependency effects between the control instances and the instances used to evaluate $AC_{\{*\}}$, we selected instances that have disjoint sets of misconceptions.

Balancing Annotation Difficulty. We generate instances of different annotation difficulties using the sets of semantically similar and dissimilar misconceptions that serve as our distractors. We randomly assign an equal number of tweet-misconception pairs to each difficulty-level ranging from very easy to very difficult. The resulting 50 instances for our final study span similar ranges in terms of length, as shown in Table 5, which is crucial to minimize the influence of reading time on our results. Overall, each of the five difficulty-levels consists of ten (two for the control instances) unique tweets that are annotated by all participants in different order.

Study Process. The final study consists of 50 instances that are ordered corresponding to the group a participant has been assigned to. Each instance consists of a tweet and six possible misconceptions (one expert-annotated and five distractors) from which the

Table 5
Average number of characters per tweet (T) and tweet and misconception (T & MC) across all difficulty-levels of annotated items.

# Chars	very easy	easy	medium	difficult	very difficult
T	219	211	183	217	194
T & MC	638	603	599	586	593

Table 6
Mean, standard deviation, and 25%, 50%, and 75% percentiles of annotation (in seconds). Σ_t denotes the total annotation time an annotator of the respective group requires to finish the study (on average).

	Σ_t	μ_t	σ_t	25%	50%	75%
Random	1,852.9	27.3	27.2	12.9	18.2	29.5
AC_{mlm}	1,273.4	23.2	19.4	**11.7**	18.6	27.4
AC_{GP}	1,324.3	26.4	19.0	14.9	20.7	30.8
AC_{gold}	**1,059.6**	**21.2**	**12.8**	12.6	**18.0**	**26.5**

participants are asked to select the most appropriate one. The lists of the six presented misconceptions are ordered randomly to prevent that participants learn to annotate a specific position. Finally, we ask each participant to answer a questionnaire that measures the perceived difficulty of the annotated instances.

5.3 General Results

In total, each of the 40 participants has provided 60 annotations, resulting in 400 annotations for the ten control instances (100 per group) and 2,000 annotations for the 50 final study instances (500 per group). In terms of annotation difficulty, each of the five difficulty-levels consists of 80 annotations for the control instances and 400 annotations for the final study. To assess the validity of $AC_{\{*\}}$, we require two criteria to be fulfilled:

H1 The participant groups do not significantly differ in terms of annotation time or annotation quality for the control instances.

H2 $AC_{\{*\}}$ shows a significant difference in annotation time or annotation quality compared to Random or each other.

Outliers. Across all 2,400 annotations, we identify only two cases where participants required more than ten minutes for annotation and are apparent outliers. To avoid removing annotations for evaluation, we compute the mean and standard deviation of the annotation time across all annotations (excluding the two outliers) and set the maximum value to $t_{max} = \mu + 5\sigma = 156.39$ seconds. This results in ten annotations that are set to t_{max} for Random, three for AC_{mlm}, one for AC_{GP}, and zero for AC_{gold}. Note that this mainly favors the random control group that serves as our baseline.

Annotation Time. Table 6 shows the results of the final study in terms of annotation time per group. Overall, annotators of AC_{gold} required on average the least amount of

Table 7
Mean, standard deviation, and 25%, 50%, and 75% percentiles of annotation quality (in percent accuracy).

	μ_{acc}	σ_{acc}	25%	50%	75%
Random	84.7	4.22	82.0	**86.0**	**88.0**
AC_{mlm}	83.6	5.32	80.0	84.0	86.0
AC_{GP}	83.6	**2.95**	82.0	**86.0**	86.0
AC_{gold}	**85.6**	3.01	**84.0**	84.0	**88.0**

time per instance and had the lowest standard deviation. We also observe a substantial decrease in the maximum annotation time, as shown in the 75th percentile for AC_{gold}. Conducting a Kruskal–Wallis test (Kruskal and Wallis 1952) on the control instances across all participant groups results in a p-value of $p = 0.200 > 0.05$.[10] Hence, we cannot reject the null-hypothesis for the control instances, and conclude that all groups initially do not show statistically significant differences in terms of annotation time for the control instances, thereby satisfying H1. Next, we conduct the same test on the evaluation instances and observe a statistically significant p-value of $p = 4.53^{-6} < 0.05$. For a more specific comparison, we further conduct pairwise Welch's t-test (Welch 1951) for each strategy with a Bonferroni-corrected p-value of $p = \frac{0.05}{6} = 0.008\overline{3}$ to account for multiple comparisons (Bonferroni 1936). Overall, AC_{gold} performs best, satisfying H2 with statistically significant improvements over Random ($p = 7.28^{-6}$) and AC_{GP} ($p = 3.79^{-7}$). Although the difference to AC_{mlm} is substantial, it is not statistically significant ($p = 0.0502$). The best performing estimator is AC_{mlm}, which performs significantly better than Random ($p = 0.0069$) and substantially better than AC_{GP} ($p = 0.0084$). Between AC_{GP} and Random, we cannot observe any statistically significant differences ($p = 0.5694$).

Annotation Quality. We evaluate annotation quality by computing the accuracy for each participant, that is, the percentage of misconceptions that they were able to correctly identify out of the six presented ones. Table 7 shows our results in terms of accuracy. Although AC_{gold} has the highest mean accuracy, the most differences lie within the range of 2% accuracy, which is equivalent to only a single wrongly annotated instance. Conducting Kruskal–Wallis tests for the control instances shows that the difference in terms of accuracy is not statistically significant ($p = 0.881$), satisfying H1. However, the same test shows no statistically significant difference for the final study ($p = 0.723$). One reason for this may be our decision to conduct the study with voluntary participants and their higher intrinsic motivation to focus on annotation quality over annotation time (Chau et al. 2020). In contrast to crowdsourcing scenarios where annotators are mainly motivated by monetary gain—trying to reduce the amount of time they spend on their annotation at the cost of quality—voluntary annotators are more motivated to invest additional time to provide correct annotations; even more so in a setup with a low number of 60 instances.

10 In general, ANOVA (analysis of variance) is a more expressive test that does not require pairwise comparisons that are necessary for the less expressive Kruskal–Wallis test. However, we cannot apply ANOVA in our case due to violated conditions on normality and homoscedasticity of the collected data.

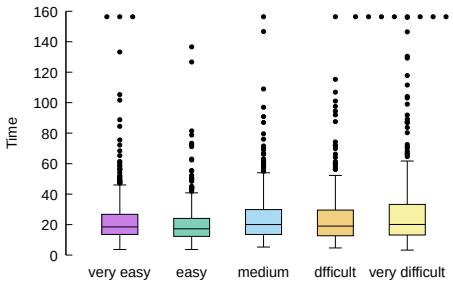


Figure 4
Annotation time (in seconds) grouped by difficulty level.

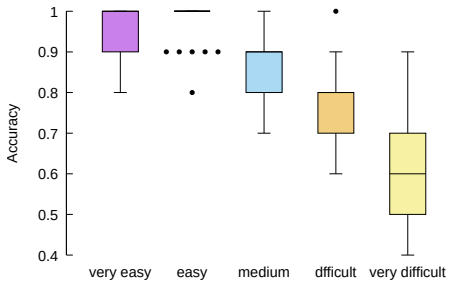


Figure 5
Accuracy per annotator grouped by difficulty level.

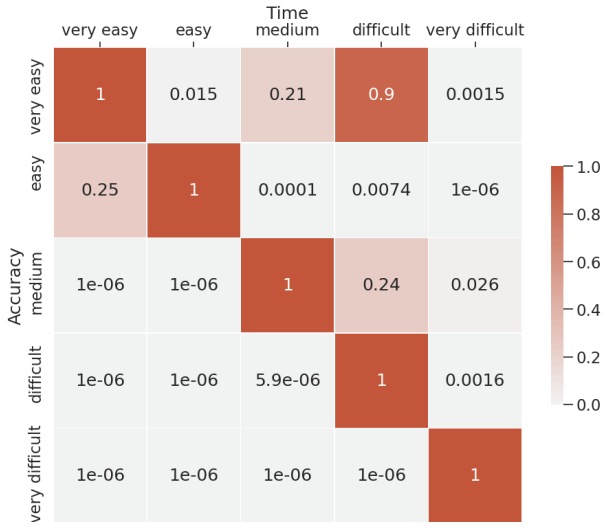


Figure 6
The p-values for time (in seconds) and accuracy between different difficulty levels.

Difficulty Evaluation. To validate our generation approach with distractors, we further evaluate all annotation instances in terms of their annotation difficulty. As Figures 4 and 5 show, one can observe non-negligible differences in terms of annotation time as well as accuracy across instances of different difficulties. Conducting pairwise Welch's t-tests with a Bonferroni corrected p-value of $p = \frac{0.05}{10} = 0.005$ shows that in terms of accuracy, only very easy and easy instances do not express a statistically significant difference ($p = 0.25$), showing that participants had more trouble in identifying the correct misconception for difficult instances.[11] For all other instances, we observe p-values smaller than $1e^{-6}$, as shown in Figure 6. In terms of annotation time, the differences are not as apparent as in annotation accuracy. We find statistically significant differences in only four out of ten cases showing that the annotation difficulty does not necessarily impact the annotation time. Overall, we still observe that instances express

11 Overall, we require $\frac{n(n-1)}{2}$ pairwise comparisons, resulting in 10 comparisons with $n = 5$.

significant differences in terms of either annotation time or quality (or both), showing that our approach using distractor sets to control the annotation difficulty worked well.

5.4 Error Analysis

Model Performance. While AC_{mlm} and AC_{gold} both outperform the random baseline significantly, AC_{GP} does not. To analyze how well the used GP model performs for individual annotators, we perform leave-one-user-out cross validation experiments across all 40 participants. Table 8 shows the MAE, RMSE, the coefficient of determination (R^2), and Spearman's ρ of our experiments. Overall, we find a low correlation between the predicted and true annotation time and high standard deviations across both errors. Further analyzing the performance of AC_{GP} for interactively predicting the annotation time (cf. Figure 7) shows that the model adapts rather slowly to additional data. As can be observed, the low performance of the model (MAE between 10 and 20 seconds) results in a high variation in the annotation time of the selected instances between subsequent iterations; further experiments strongly suggest this is due to the model suffering from a cold start and the small amount of available training data as also discussed below.

Correlation with AC_{gold}. A second shortcoming of AC_{GP} becomes apparent when observing the difficulty of the sampled instances across all iterations, shown in Figure 8. We observe a low Spearman's ρ correlation to AC_{gold} of 0.005, in contrast to AC_{mlm} ($\rho = 0.22$). Only Random has a lower correlation, of $\rho = -0.15$. This shows that model adaptivity plays an important role, especially in low-data scenarios such as in early

Table 8
Leave-one-out cross validation results on annotation times, grouped by user and averaged.

	μ_t	σ_t	25%	50%	75%
MAE	12.4	6.1	8.5	10.4	14.3
RMSE	17.2	9.1	11.1	13.9	20.3
R^2	0.0	0.0	−0.1	0.0	0.0
ρ	−0.1	0.2	−0.3	−0.1	0.1

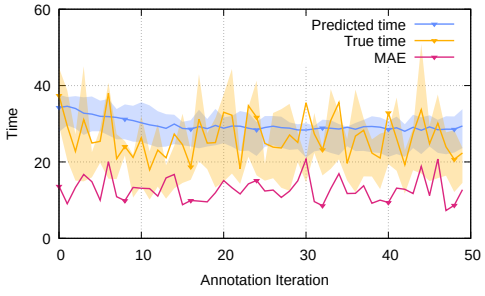


Figure 7
Mean, lower, and upper percentiles for predicted and true annotation time and the mean absolute error.

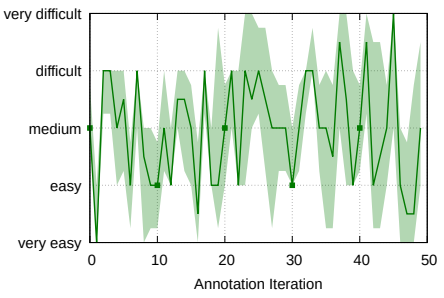


Figure 8
Median, lower, and upper percentile for instance difficulty with AC_{GP} at each iteration.

Table 9
Spearman's ρ correlation analysis for three potential confounding factors.

	CEFR		Annotator		Conductor	
	ρ	p-value	ρ	p-value	ρ	p-value
Time	−0.307	0.054	−0.134	0.409	0.085	0.600
Accuracy	0.319	0.044	−0.060	0.711	−0.211	0.191

stages during annotation studies. We plan to tackle this issue in future work using more sophisticated models and combined approaches that initially utilize heuristics and switch to interactively trained models with the availability of sufficient training data.

5.5 Participant Questionnaire

After completing the annotation study, each participant answered a questionnaire quantifying their language proficiency, previous annotation experience, and perceived difficulty of the annotation task.

Language Proficiency. In addition to their CEFR language proficiency (Council of Europe. 2001), we further asked participants to provide optional information about their first language and the number of years they have been actively speaking English. On average, our participants have been actively speaking English for more than 10 years. Overall, they stated a language proficiency of: B1 (1), B2 (11), C1 (17), and C2 (11). Most of our participants stated German as their first language (30). Other first languages include Vietnamese (4), Chinese (3), Russian (1), and Czech (1).[12]

Annotation Experience. We further collected data from our participants regarding their previous experience as study participants as well as study conductors. In general, about 50% of our participants (18) had not participated in annotation studies before. Nineteen had participated in a few (one to three) studies, and only three in more than three studies. Even more participants had not previously conducted a study (24) or only a few (12). In total, four participants stated that they had set up more than three annotation studies.

Confounding Factors. We identify the language proficiency and previous experience with annotation studies as potential confounding factors (VanderWeele and Shpitser 2013). Confounding factors are variables that are difficult to control for, but have an influence on the whole study and can lead to a misinterpretation of the results. Especially in studies that include a randomized setup such as in ours—due to the random assignment of our participants into the four groups—it is crucial to investigate the influence of potential confounding factors. In our analysis, we focus on variables for which all participants provided an answer, namely, their CEFR level and their experience as participants in and conductors of annotation studies (some of our participants were researchers). Table 9 shows the results of a Spearman's ρ correlation analysis for all three variables against annotation time and accuracy. As we can see, the participants'

12 One participant decided not to disclose any additional information except English proficiency.

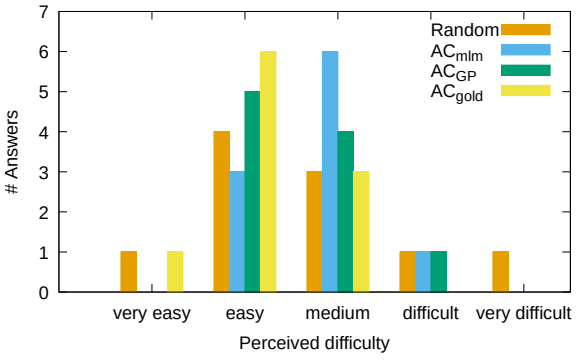


Figure 9
Accumulated perceived difficulty answers across all groups.

experiences as annotators (Annotator) or study conductors (Conductor) only yields a low, non-significant correlation with time and accuracy and, consequently, can be excluded as confounding factors. The influence of their language proficiency (CEFR) is more interesting, as it shows a small negative correlation for annotation time and a small positive correlation for annotation accuracy with p-values around 0.05, meaning that participants with a lower CEFR level required less time, but also had a lower accuracy. To investigate the influence of a participant's language proficiency on our results, we conduct a Kruskal–Wallis test for the distribution of different language proficiency levels across the four groups and find that they do not differ significantly with a p-value of $p = 0.961$. Nonetheless, we find that the CEFR level is an important confounding factor that needs to be considered in future study setups.

Perceived Difficulty. To quantify if there exists any difference between the actual difficulty and the perceived difficulty, we further asked our participants the following questions:

PQ1: How difficult did you find the overall annotation task?

PQ2: Did you notice any differences in difficulty between individual tweets?

PQ3: Would you have preferred a different ordering of the items?

Figure 9 shows the distribution of answers (from very easy to very difficult) to PQ1 across all four groups. Interestingly, whereas participants of the AC_{mlm} group did require less time during their annotation compared with AC_{GP}, more people rated the study as of medium difficulty than participants of AC_{GP}. This may be an indicator that AC_{GP} may—although not measurable in terms of annotation time—alleviate the perceived difficulty for participants, hence, still reducing the cognitive burden. We will investigate this in further studies that also include an item specific difficulty annotation, that is, by explicitly asking annotators for the perceived difficulty.[13] Overall, only four out of 40 participants (two for AC_{GP} and one for AC_{mlm} and AC_{gold} each) did state to not have noticed any differences in terms of difficulty between different instances; showing that the selected distractors resulted in instances of noticeably different annotation

13 We excluded this additional annotation in the study as one pass already required ∼ 45–60 minutes.

difficulty (PQ2). For PQ3, we find that 33 participants did not wish for a different ordering of instances (but were still allowed to provide suggestions), four would have preferred an "easy-first," one a "difficult-first," and two an entirely different ordering strategy. From the 14 free-text answers and feedback via other channels, we identify three general suggestions that may be interesting for future research:

S1: Grouping by word rarity.

S2: Grouping instances by token overlap.

S3: Grouping instances by topic (tweet or alternatively, misconception) similarity.

Further analyzing the free-text answers together with the pre-defined answers ("no," "easy-first," "difficult-first," and "other") shows that the participants disagree on the preferred ordering strategy. For instance, the participants that suggested S3, disagreed if instances should be grouped by topic similarity to reduce the number of context switches or be as diverse as possible to provide some variety during annotation. Another five participants (two from Random and one from the other groups each) even explicitly supported a random ordering in the free-text answer. The disagreement upon the ordering strategy shows the importance of interactively trained estimators that are capable of providing personalized annotation curricula.

6. Limitations and Future Work

We evaluated AC with an easy-instances-first strategy in simulations as well as in a highly controlled setup using a finite, pre-annotated data set and task-agnostic estimators to minimize possible noise factors. To demonstrate the viability of AC with a sufficient number of voluntary annotators, we further chose a dataset that covers a widely discussed topic and manually controlled the annotation difficulty to make it accessible for non-experts. To evaluate AC with more generalizable results in a real-world scenario, we discuss existing limitations that should be considered beforehand that can also serve as promising research directions for future work.

Difficulty Estimators. Due to novelty of the proposed approach and the lack of well-established baselines, we focused on task-agnostic annotation difficulty estimators such as reading difficulty and annotation time, which can easily be applied to a wide range of tasks. Although our study results show that they work to some extent, our evaluation with existing datasets also shows that especially non-adaptive estimators, which approximate the absolute task-difficulty, are sensitive to the data domain and annotation task (cf. the low performance of length-based estimators on the SigIE data in Section 4). Such issues could be addressed by implementing estimators that are more *task-specific*. For named entity annotations, a general improvement may be achieved by considering the number of nouns within a sentence that can be obtained from a pre-trained part-of-speech tagger. One may even consider domain-specific word frequency lists to provide a difficulty estimate for entities. For instance, among the annotated named entities in Muc7_{T}, "U.S." (occurs 72 times) may be easier to annotate than "Morningstar" (occurs only once); simply based on a word frequency analysis. Other, more sophisticated approaches from educational research such as item response theory (Baker 2001) and scaffolding (Jackson et al. 2020) may also lead to better task-agnostic estimators. Such

approaches and combinations of task-agnostic with task-specific estimators remain to be investigated in future work.

Annotation Strategies. In this work, we focused on developing and evaluating a strategy for our non-expert annotation scenario. Although it proved to be effective in our user study, we also find that our annotators disagree in their preferences with respect to the ordering of instances—which indicates that investigating *annotator-specific* strategies could be a promising line for future work. Another shortcoming of the evaluated strategy is that it does not consider an annotator's boredom or frustration (Vygotsky 1978). Especially when considering larger annotation studies, motivation may become an increasingly important factor with non-expert annotators as they further progress in a task and become more proficient. Such a strategy may also be better suited for annotation scenarios that involve domain experts to retain a high motivation by avoiding boredom—for instance, by presenting them with subsequent instances of varying difficulty or different topics. Domain experts who do not require a task-specific training may also benefit from strategies that focus on familiarizing them with the data domain early on to provide them with a good idea of what kind of instances they can expect throughout their annotations. To implement strategies that consider annotator-specific factors such as motivation and perceived difficulty, adaptive estimators may have an advantage over non-adaptive ones as they can incorporate an annotator's preference on the fly. We will investigate more sophisticated adaptive estimators (also coupled with non-adaptive ones) and strategies in future work and also plan to evaluate AC with domain expert annotators.

Larger Datasets. While using a finite set of annotated instances was necessary in our user study to ensure a proper comparability, AC is not limited to annotation scenarios with finite sets. However, deploying AC in scenarios that involve a large number of unlabeled instances requires additional consideration besides an annotator's motivation. In scenarios that only annotate a subset of the unlabeled data (similar to pool-based active learning), an easy-instances-first strategy may lead to a dataset that is imbalanced toward instances that are easy to annotate. This can hurt data diversity and consequently result in models that do not generalize well to more difficult instances. To create more diverse datasets, one may consider introducing a stopping criterion (e.g., a fixed threshold) for the annotator training phase and moving on to a different sampling strategy from active learning. Other, more sophisticated approaches would be to utilize adaptive estimators with a pacing function (Kumar, Packer, and Koller 2010) or sampling objectives that jointly consider annotator training and data diversity (Lee, Meyer, and Gurevych 2020). Such approaches are capable of monitoring the study progress and can react accordingly, which may result in more diverse datasets. However, they also face additional limitations in terms of the computational overhead that may require researchers to consider an asynchronous model training in their setup.

Implementation Overhead. Finally, to apply AC in real-world annotation studies, one needs to consider the additional effort for study conductors to implement it. Whereas the task-agnostic estimators we provide can be integrated with minimal effort, developing task- and annotator-specific estimators may not be a trivial task and requires a profound knowledge about the task, data, and annotators. Another open question is how well the time saving of approximately 8–13 minutes per annotator in our study translates to large-scale annotation studies. If so, then AC could also be helpful in annotation studies with domain experts by resulting in more annotated instances within

a fixed amount of time—however, if not, this would simply lead to a trade-off between the time investment of the study conductor and annotators. Overall, we find that developing and evaluating further strategies and estimators to provide study conductors with a wide range of choices to consider for their annotation study will be an interesting task for the research community.

7. Conclusion

With annotation curricula, we have introduced a novel approach for implicitly training annotators. We provided a formalization for an easy-instances-first strategy that orders instances from easy to difficult by approximating the annotation difficulty with task-agnostic heuristics and annotation time. In our experiments with three English datasets, we identified well-performing heuristics and interactively trained models and find that the data domain and the annotation task can play an important role when creating an annotation curriculum. Finally, we evaluate the best performing heuristic and adaptive model in a user study with 40 voluntary participants who classified English tweets about the Covid-19 pandemic and show that leveraging AC can lead to a significant reduction in annotation time while preserving annotation quality.

With respect to our initial research questions (cf. Section 1), our results show that the order in which instances are annotated can have a statistically significant impact in terms of annotation time (RQ1) and that recent language models can provide a strong baseline to pre-compute a well-performing ordering (RQ2). We further find that our interactively trained regression models lack adaptivity (RQ3), as they perform well on existing datasets with hundreds or more training instances, but fall behind non-adaptive estimators in the user study.

We conclude that annotation curricula provide a promising way for more efficient data acquisition in various annotation scenarios—but that they also need further investigation with respect to task-specific estimators for annotation difficulty, annotator-specific preferences, and applicability on larger datasets. Our analysis of existing work shows that, unfortunately, the annotation ordering as well as annotation times are seldomly reported. In the face of the increasing use of AI models in high-stake domains (Sambasivan et al. 2021) and the potentially harmful impact of biased data (Papakyriakopoulos et al. 2020), we ask dataset creators to consider including individual annotation times and orderings along with a datasheet (Gebru et al. 2021) when publishing their dataset. To facilitate future research, we share all code and data and provide a ready-to-use and extensible implementation of AC in the INCEpTION annotation platform.[14]

Acknowledgments

This work has been supported by the European Regional Development Fund (ERDF) and the Hessian State Chancellery – Hessian Minister of Digital Strategy and Development under the promotional reference 20005482 (TexPrax) and the German Research Foundation under grant nos. EC 503/1-1 and GU 798/21-1 (INCEpTION). We thank Michael Bugert, Richard Eckart de Castilho, Max Glockner, Ulf Hamster, Yevgeniy Puzikov, Kevin Stowe, and the anonymous reviewers for their thoughtful comments and feedback, as well as all anonymous participants in our user study.

14 https://inception-project.github.io/.

Ethics Statement

Informed Consent. Participants of our user study participated voluntarily and anonymously with a self-chosen, unique study key that allows them to request the deletion of their provided data at a later point. Upon registration, they are informed about the data presented and collected in the study, its further use, and the purpose of the study. Before collecting any data, participants are explicitly asked for their informed consent. We do not collect any personal data in our study. If participants do not provide their informed consent, their study key is deleted immediately. For publication, the study key is further replaced with a randomly generated user id.

Use of Twitter Data. The CovidLies corpus (Hossain et al. 2020) we used to generate the instances for our annotation study consists of annotated tweets. To protect the anonymity of the user who created the tweet, we only display the text (removing any links) without any metadata like Twitter user id or timestamps to our study participants. We only publish the tweet ids in our study data to conform with Twitter's terms of service and hence, all users retain their right to delete their data at any point.

References

Agarwal, Manish and Prashanth Mannem. 2011. Automatic gap-fill question generation from text books. In *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 56–64.

Ash, Jordan T., Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, pages 1–26.

Baker, Frank. 2001. *The Basics of Item Response Theory*. ERIC Clearinghouse on Assessment and Evaluation, College Park, MD.

Beck, Tilman, Ji-Ung Lee, Christina Viehmann, Marcus Maurer, Oliver Quiring, and Iryna Gurevych. 2021. Investigating label suggestions for opinion mining in German Covid-19 social media. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–13. https://doi.org/10.18653/v1/2021.acl-long.1

Beigman Klebanov, Beata and Eyal Beigman. 2014. Difficult cases: From data to learning, and back. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 390–396. https://doi.org/10.3115/v1/P14-2064

Beinborn, Lisa, Torsten Zesch, and Iryna Gurevych. 2014. Predicting the difficulty of language proficiency tests. *Transactions of the Association for Computational Linguistics*, 2:517–529. https://doi.org/10.1162/tacl_a_00200

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48. https://doi.org/10.1145/1553374.1553380

Bonferroni, Carlo. 1936. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62.

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. https://doi.org/10.18653/v1/D15-1075

Chau, Hung, Saeid Balaneshin, Kai Liu, and Ondrej Linda. 2020. Understanding the tradeoff between cost and quality of expert annotations for keyphrase extraction. In *Proceedings of the 14th Linguistic Annotation Workshop*, pages 74–86.

Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Council for Cultural Co-operation. Education Committee. Modern Languages Division. Cambridge University Press, Strasbourg, France.

Deutsch, Tovly, Masoud Jasbi, and Stuart Shieber. 2020. "Linguistic features for

readability assessment." In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–17. https://doi.org/10.18653/v1/2020.bea-1.1

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Fang, Meng, Jie Yin, and Dacheng Tao. 2014. Active learning for crowdsourcing using knowledge transfer. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1809–1815.

Fang, Meng, Xingquan Zhu, Bin Li, Wei Ding, and Xindong Wu. 2012. Self-taught active learning from crowds. In *2012 IEEE 12th International Conference on Data Mining*, pages 858–863. https://doi.org/10.1109/ICDM.2012.64

Felice, Mariano and Paula Buttery. 2019. Entropy as a proxy for gap complexity in open cloze tests. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 323–327. https://doi.org/10.26615/978-954-452-056-4_037

Fort, Karën and Benoît Sagot. 2010. Influence of pre-annotation on POS-tagged corpus development. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 56–63.

Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1923–1932.

Gao, Lingyu, Kevin Gimpel, and Arnar Jensson. 2020. Distractor analysis and selection for multiple-choice cloze questions for second-language learners. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 102–114.

Gebru, Timnit, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Communications of ACM*, 64(12):86–92. https://doi.org/10.1145/3458723

Geva, Mor, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? An investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166. https://doi.org/10.18653/v1/D19-1107

Gururangan, Suchin, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360. https://doi.org/10.18653/v1/2020.acl-main.740

Hossain, Tamanna, Robert L. Logan IV, Arjuna Ugarte, Yoshitomo Matsubara, Sean Young, and Sameer Singh. 2020. COVIDLies: Detecting COVID-19 misinformation on social media. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, pages 1–11. https://doi.org/10.18653/v1/2020.nlpcovid19-2.11

Huang, Sheng-Jun, Rong Jin, and Zhi-Hua Zhou. 2010. Active learning by querying informative and representative examples. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, pages 892–900.

Jackson, Corey, Carsten Østerlund, Kevin Crowston, Mahboobeh Harandi, Sarah Allen, Sara Bahaadini, Scotty Coughlin, Vicky Kalogera, Aggelos Katsaggelos, Shane Larson, et al. 2020. Teaching citizen scientists to categorize glitches using machine learning guided training. *Computers in Human Behavior*, 105:1–11. https://doi.org/10.1016/j.chb.2019.106198

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3149–3157.

Kelly, A. V. 2009. *The Curriculum: Theory and Practice*, SAGE Publications, Thousand Oaks, CA.

Kicikoglu, Osman Doruk, Richard Bartle, Jon Chamberlain, Silviu Paun, and Massimo Poesio. 2020. Aggregation driven progression system for GWAPs. In *Workshop on Games and Natural Language Processing*, pages 79–84.

Kincaid, J. Peter, Robert P. Fishburne Jr., Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability

formulas (automated readability index, fog count and Flesch reading ease formula) for Navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch. https://doi.org/10.21236/ADA006655

Kirsch, Andreas, Joost van Amersfoort, and Yarin Gal. 2019. BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7026–7037.

Klie, Jan Christoph, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9.

Klie, Jan Christoph, Richard Eckart de Castilho, and Iryna Gurevych. 2020. From zero to hero: Human-in-the-loop entity linking in low resource domains. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6982–6993. https://doi.org/10.18653/v1/2020.acl-main.624

Krashen, Stephen. 1982. *Principles and Practice in Second Language Acquisition*, Pergamon Press, Oxford and New York.

Kruskal, William H. and W. Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621. https://doi.org/10.1080/01621459.1952.10483441

Kumar, M., Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23, pages 1189–1197.

Laws, Florian, Christian Scheible, and Hinrich Schütze. 2011. Active learning with Amazon Mechanical Turk. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556.

Lee, Ji Ung, Christian M. Meyer, and Iryna Gurevych. 2020. Empowering active learning to jointly optimize system and user demands. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4233–4247.

Lee, Ji Ung, Erik Schwan, and Christian M. Meyer. 2019. Manipulating the difficulty of c-tests. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 360–370. https://doi.org/10.18653/v1/P19-1035

Lewis, David D. and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. https://doi.org/10.1007/978-1-4471-2099-5_1

Lingren, Todd, Louise Deleger, Katalin Molnar, Haijun Zhai, Jareen Meinzen-Derr, Megan Kaiser, Laura Stoutenborough, Qi Li, and Imre Solti. 2014. Evaluating the impact of pre-annotation on annotation speed and potential bias: Natural language processing gold standard development for clinical named entity recognition in clinical trial announcements. *Journal of the American Medical Informatics Association*, 21(3):406–413. https://doi.org/10.1136/amiajnl-2013-001837, PubMed: 24001514

Loukina, Anastassia, Su-Youn Yoon, Jennifer Sakano, Youhua Wei, and Kathy Sheehan. 2016. Textual complexity as a predictor of difficulty of listening items in language proficiency tests. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3245–3253.

Lowell, David, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 21–30. https://doi.org/10.18653/v1/D19-1003

Madge, Chris, Juntao Yu, Jon Chamberlain, Udo Kruschwitz, Silviu Paun, and Massimo Poesio. 2019. Progression in a language annotation game with a purpose. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, pages 77–85.

Martínez Alonso, Héctor, Barbara Plank, Anders Johannsen, and Anders Søgaard. 2015. Active learning for sense annotation. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 245–249.

Mayfield, Elijah and Alan W. Black. 2020. Should you fine-tune BERT for automated essay scoring? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*,

pages 151–162. https://doi.org/10.18653/v1/2020.bea-1.15

Mostow, Jack and Hyeju Jang. 2012. Generating diagnostic multiple choice comprehension cloze questions. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 136–146.

Nguyen, Hieu T. and Arnold Smeulders. 2004. Active learning using pre-clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 79–87. https://doi.org/10.1145/1015330.1015349

Nie, Yixin, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901. https://doi.org/10.18653/v1/2020.acl-main.441

Pak, Alexander and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 1320–1326.

Papakyriakopoulos, Orestis, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. 2020. Bias in word embeddings. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 446–457. https://doi.org/10.1145/3351095.3372843

Paun, Silviu, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing Bayesian models of annotation. *Transactions of the Association for Computational Linguistics*, 6:571–585. https://doi.org/10.1162/tacl_a_00040

Pedregosa, Fabian, 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. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Peters, Matthew E., Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14. https://doi.org/10.18653/v1/W19-4302

Reimers, Nils and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. https://doi.org/10.18653/v1/D19-1410

Rogers, Anna. 2021. Changing the world by changing the data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2182–2194. https://doi.org/10.18653/v1/2021.acl-long.170

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

Sakaguchi, Keisuke, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An adversarial Winograd Schema Challenge at scale. *Communications of ACM*, 64(9):99–106. https://doi.org/10.1145/3474381

Salazar, Julian, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712. https://doi.org/10.18653/v1/2020.acl-main.240

Sambasivan, Nithya, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M. Aroyo. 2021. "Everyone wants to do the model work, not the data work": Data cascades in high-stakes AI. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, pages 1–15. https://doi.org/10.1145/3411764.3445518

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Schulz, Claudia, Christian M. Meyer, Jan Kiesewetter, Michael Sailer, Elisabeth Bauer, Martin R. Fischer, Frank Fischer, and Iryna Gurevych. 2019. Analysis of automatic annotation suggestions for hard discourse-level tasks in expert domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2761–2772. https://doi.org/10.18653/v1/P19-1265

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. https://doi.org/10.18653/v1/P16-1162

Settles, Burr. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114. https://doi.org/10.2200/S00429ED1V01Y201207AIM018

Settles, Burr, Mark Craven, and Lewis Friedland. 2008. Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, pages 1–10.

Siddhant, Aditya and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909. https://doi.org/10.18653/v1/D18-1318

Snow, Rion, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263. https://doi.org/10.3115/1613715.1613751

Stab, Christian, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018. ArgumenText: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 21–25. https://doi.org/10.18653/v1/N18-5005

Sweetser, Penelope and Peta Wyeth. 2005. GameFlow: A model for evaluating player enjoyment in games. *Computers in Entertainment*, 3(3):3. https://doi.org/10.1145/1077246.1077253

Szpiro, George. 2010. *Numbers Rule: The Vexing Mathematics of Democracy, from Plato to the Present*. Princeton University Press. https://doi.org/10.1515/9781400834440

Tauchmann, Christopher, Johannes Daxenberger, and Margot Mieskes. 2020. The influence of input data complexity on crowdsourcing quality. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion*, pages 71–72. https://doi.org/10.1145/3379336.3381499

Taylor, Wilson L. 1953. "Cloze procedure": A new tool for measuring readability. *Journalism & Mass Communication Quarterly*, 30(4):415–433. https://doi.org/10.1177/107769905303000401

Tomanek, Katrin and Udo Hahn. 2009. Timed annotations: Enhancing MUC7 metadata by the time it takes to annotate named entities. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 112–115. https://doi.org/10.3115/1698381.1698399

Turner, Brandon M. and Dan R. Schley. 2016. The anchor integration model: A descriptive model of anchoring effects. *Cognitive Psychology*, 90:1–47. https://doi.org/10.1016/j.cogpsych.2016.07.003, PubMed: 27567237

VanderWeele, Tyler J. and Ilya Shpitser. 2013. On the definition of a confounder. *Annals of Statistics*, 41(1):196–220. https://doi.org/10.1214/12-AOS1058, PubMed: 25544784

Vygotsky, Lev. 1978. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press.

Wang, Min, Fan Min, Zhi-Heng Zhang, and Yan-Xue Wu. 2017. Active learning through density clustering. *Expert Systems with Applications*, 85:305–317. https://doi.org/10.1016/j.eswa.2017.05.046

Welch, Bernard Lewis. 1951. On the comparison of several mean values: An alternative approach. *Biometrika*, 38(3/4):330–336. https://doi.org/10.1093/biomet/38.3-4.330

Xia, Menglin, Ekaterina Kochmar, and Ted Briscoe. 2016. "Text readability assessment

for second language learners." In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22. https://doi.org/10.18653/v1/W16-0502

Yang, Yinfei, Oshin Agarwal, Chris Tar, Byron C. Wallace, and Ani Nenkova. 2019. Predicting annotation difficulty to improve task routing and model performance for biomedical information extraction. 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 1471–1480. https://doi.org/10.18653/v1/N19-1150

Yeung, Chak Yan, John Lee, and Benjamin Tsou. 2019. Difficulty-aware distractor generation for gap-fill items. In *Proceedings of the 17th Annual Workshop of the Australasian Language Technology Association*, pages 159–164.

Yimam, Seid Muhie, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. 2014. Automatic annotation suggestions and custom annotation layers in WebAnno. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96. https://doi.org/10.3115/v1/P14-5016

Yuan, Michelle, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. 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. https://doi.org/10.18653/v1/2020.emnlp-main.637

Zhang, Chicheng and Kamalika Chaudhuri. 2015. Active learning from weak and strong labelers. In *Advances in Neural Information Processing Systems*, pages 703–711.

Zhu, Jingbo, Huizhen Wang, Tianshun Yao, and Benjamin K. Tsou. 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1137–1144. https://doi.org/10.3115/1599081.1599224