Analysis of Automatic Annotation Suggestions for Hard Discourse-Level Tasks in Expert Domains

Claudia Schulz[1], **Christian M. Meyer**[1], **Jan Kiesewetter**[2], **Michael Sailer**[3],
Elisabeth Bauer[3], **Martin R. Fischer**[2], **Frank Fischer**[3], **and Iryna Gurevych**[1]
[1] Ubiquitous Knowledge Processing (UKP) Lab, Technische Universität Darmstadt, Germany
[2] Institute of Medical Education, University Hospital, LMU München, Germany
[3] Chair of Education and Educational Psychology, LMU München, Germany
http://famulus-project.de

Abstract

Many complex discourse-level tasks can aid domain experts in their work but require costly expert annotations for data creation. To speed up and ease annotations, we investigate the viability of automatically generated annotation suggestions for such tasks. As an example, we choose a task that is particularly hard for both humans and machines: the segmentation and classification of epistemic activities in diagnostic reasoning texts. We create and publish a new dataset covering two domains and carefully analyse the suggested annotations. We find that suggestions have positive effects on annotation speed and performance, while not introducing noteworthy biases. Envisioning suggestion models that improve with newly annotated texts, we contrast methods for continuous model adjustment and suggest the most effective setup for suggestions in future expert tasks.

1 Introduction

Current deep learning methods require large amounts of training data to achieve reasonable performance. Scalable solutions to acquire labelled data use crowdsourcing (e.g., Potthast et al., 2018), gamification (Ahn, 2006), or incidental supervision (Roth, 2017). For many complex tasks in expert domains, such as law or medicine, this is, however, not an option since crowdworkers and gamers lack the necessary expertise. Annotating data manually is therefore often the only way to train a model for tasks aiding experts with their work. But the more expertise an annotation task requires, the more time- and funding-intensive it typically is, which is why many projects suffer from small corpora and deficient models.

In this paper, we propose and analyse an annotation setup aiming to increase the annotation speed and ease for a discourse-level sequence labelling task requiring extensive domain expertise, without sacrificing annotation quality. For the first time, we study the effects of automatically suggesting annotations to expert annotators in a task that is hard for both humans (only moderate agreement) and machine learning models (only mediocre performance) and compare the effects across different domains and suggestion models. We furthermore investigate how the performance of the models changes if they continuously learn from expert annotations.

As our use case, we consider the task of annotating epistemic activities in diagnostic reasoning texts, which was recently introduced by Schulz et al. (2018, 2019). The task is theoretically grounded in the learning sciences (Fischer et al., 2014) and enables innovative applications that teach diagnostic skills to university students based on automatically generated feedback about their reasoning processes. This task is an ideal choice for our investigations, since it is novel, with limited resources and experts available, and so far neural prediction models only achieve an F1 score of 0.6, while also human agreement is in a mid range around $\alpha = 0.65$.

Schulz et al. (2018) created annotated corpora of epistemic activities for 650 texts in the medicine domain (MeD) and 550 in the school teaching domain (TeD). We extend these corpora by 457 and 394 texts, respectively. As a novel component, half of the domain expert annotators receive automatically generated annotation suggestions. That is, the annotation interface features texts with (suggested) annotations rather than raw texts. Annotators can accept or reject the suggested annotations as well as add new ones, as in the standard annotation setup.

Based on the collected data, we investigate the effects of these suggestions in terms of inter- and intra-annotator agreement, annotation time, sug-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2761–2772
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

gestion usefulness, annotation bias, and the type of suggestion model. As our analysis reveals positive effects, we additionally investigate training suggestion models that learn continuously as new data becomes available. Such incremental models can benefit tasks with no or little available data.

Our work is an important step towards our vision that even hard annotation tasks in expert domains, requiring extensive training and discourse-level context, can be annotated more efficiently, thus advancing applications that aid domain experts in their work. Besides epistemic activities, discourse-level expert annotation tasks concern, for example, legal documents (Nazarenko et al., 2018), psychiatric patient–therapist interactions (Mieskes and Stiegelmayr, 2018), or transcripts of police body cameras (Voigt et al., 2017).

The contributions of our work are: (1) We study the effects of automatically suggesting annotations to expert annotators across two domains for a hard discourse-level sequence labelling task. (2) We learn incremental suggestion models for little data scenarios through continuous adjustments of the suggestion model and discuss suitable setups. (3) We publish new diagnostic reasoning corpora for two domains annotated with epistemic activities.[1]

2 Related Work

Annotation Suggestions Previous work on automatic annotation suggestion (sometimes called pre-annotation) focused on token- or sentence-level annotations, including the annotation of part-of-speech tags (Fort and Sagot, 2010), syntactic parse trees in historical texts (Eckhoff and Berdicevskis, 2016), and morphological analysis (Felt et al., 2014). A notable speed-up of the annotation could be observed in these tasks, up to 70 % (Felt et al., 2014). However, Fort and Sagot (2010) find that annotation suggestions also biased the annotators' decisions. Rosset et al. (2013) instead report no clear bias effects for their pre-annotation study of named entities. Ulinski et al. (2016) investigate the effects of different suggestion models for dependency parsing. They find that models with an accuracy of at least 55 % reduce annotation time. Our work focuses on a different class of tasks, namely hard discourse-level tasks, in which the expert annotators only achieve a moderate agreement.

Annotation tasks in the medical domain are related to our use case in diagnostic reasoning. Lingren et al. (2014) suggest medical entity annotations in clinical trial announcements. Kholghi et al. (2017) also investigate medical entity annotation, using active learning for their suggestion model, which results in a speed-up of annotation time. South et al. (2014) use automatic suggestions for de-identification of medical texts and find no change in inter-annotator agreement or annotation time. In contrast to these works, we use a control group of two annotators, who never receive suggestions, and compare the performance of all annotators to previous annotations they performed without annotation suggestions.

Work on the technical implementation of annotation suggestions is also still focused on word- or sentence-level annotation types. Meurs et al. (2011) use the GATE annotation framework (Bontcheva et al., 2013) for suggestions of biological entities. Yimam et al. (2014) describe the WebAnno system and discuss suggestions of part-of-speech tags and named entities using the MIRA algorithm (Crammer and Singer, 2003) for suggestion generation. Skeppstedt et al. (2016) introduce the PAL annotation tool, which provides suggestions and active learning for entities and chunks generated by logistic regression and SVMs. Greinacher and Horn (2018) present the annotation platform DALPHI, suggesting named entity annotations based on a recurrent neural network. Documents to be annotated are chosen by means of active learning, enabling continuous updates of the suggestion model during the annotation process. We also investigate continuous updates of the suggestion model during the annotation process, but focus on a task in which annotators require vast training and domain expertise.

Continuous Model Adjustment Read et al. (2012) distinguish two ways of training a model when new data becomes available continuously: using *batches* or *single* data points for the continuous adjustment, the latter often being referred to as online learning. We experiment with both adjustment strategies. Pérez-Sánchez et al. (2010) propose *incrementally* adjusting a neural network as new data becomes available, i.e. only using the newly available data for the update. In addition to using incremental training, we also experiment with *cumulative* training, where both previously available and new data is used for the model

[1] https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2001

> *The patient reports to be lethargic and feverish*. *From the anamnesis I learned that he had purulent tonsilitis and is still suffering from symptoms*. I first performed some laboratory tests and *notice the decreased number of lymphocytes*, which can be indicative of a bone marrow disease or an HIV infection. *The HIV test is positive*. However, the results from the blood cultures are negative, so it is a virus, parasite, or a fungal infection causing the symptoms.

Figure 1: Exemplary diagnostic reasoning text from the medicine domain, annotated with epistemic activity segments: evidence generation, *evidence evaluation*, drawing conclusions, hypothesis generation.

adjustment. Andrade et al. (2017), Castro et al. (2018), and Rusu et al. (2016) investigate adapting neural networks to new data with additional classes or even new tasks, requiring to change the structure of the neural network. Our setting is less complex as the neural network is trained on all possible classes from the beginning. Recent work also investigates pre-training neural networks before training them on the actual data (Garg et al., 2018; Shimizu et al., 2018; Serban et al., 2016). The model is thus adapted only once instead of continuously as in our work.

3 Diagnostic Reasoning Task

The annotation task proposed by Schulz et al. (2018) has interesting properties for studying the effects of annotation suggestions in hard expert tasks: (1) A small set of annotated data is available for two different domains. (2) Other than in well-understood low-level tasks, such as part-of-speech tagging or named entity recognition, the expert annotators require the discourse context to identify epistemic activities. This is a hard task yielding only inter-rater agreement scores in a mid range. (3) Prediction models only achieve F1 scores of around 0.6, which makes it unclear if the suggestion quality is sufficient.

The previously annotated data consists of 650 German texts in the medical domain (MeD) and 550 texts in the teacher domain (TeD). The texts were written by university students working on online case simulations, in which they had to diagnose the disease of a fictional patient (MeD), or the cause of behavioural problems of a fictional pupil (TeD) based on dialogues, observations, and test results. For each case simulation, the students explained their diagnostic reasoning process in a brief self-explanation text.[2]

Five (MeD) and four (TeD) domain experts annotated individual reasoning steps in the anonymised texts in terms of the epistemic activities (Fischer et al., 2014), i.e. activities involved in reasoning to develop a solution to a problem in a professional context. We focus on the four most frequently used epistemic activities for our annotations: hypothesis generation (HG), evidence generation (EG), evidence evaluation (EE), and drawing conclusions (DC). HG is the derivation of possible diagnoses, which initiates the reasoning process. EG constitutes explicit statements of obtaining evidence from information given in the case simulation or of recalling own knowledge. EE is the mentioning of evidence considered relevant for diagnosis. Lastly, DC is defined as the derivation of a final diagnosis, which concludes the reasoning process.

As shown in Figure 1, the annotation of these epistemic activities is framed as a joint discourse-level segmentation and classification task, that is, epistemic activities are segments of arbitrary length not bound to phrase or sentence level.

4 Annotating with Suggestions

To conduct our experiments with annotation suggestions, we use the same annotation task and platform as Schulz et al. (2018). We obtained their original data as well as further anonymised reasoning texts and we asked their expert annotators to participate in our annotation experiments. This allows us to study the effects of annotating data with and without suggestions, without having to account for changes in annotation performance due to individual expert annotators.

In total, we annotate 457 (MeD) and 394 (TeD) new reasoning texts during our experiments. Figure 2 shows an overview of our annotation phases in MeD with the five expert annotators A1 to A5. S1 and S2 indicate the previous annotation phases by Schulz et al. (2018). In their work, all experts first annotated the same texts (S1) and then a different set of texts each (S2). In our work, all experts annotate the same texts in all phases. We provide annotation suggestions to annotators A1

[2] Study approved by the university's ethics commission.

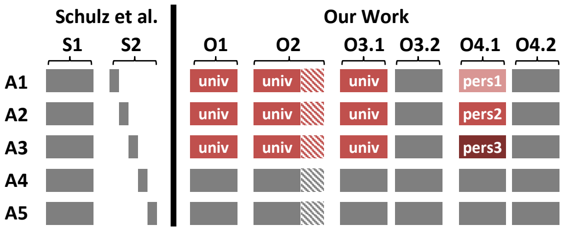


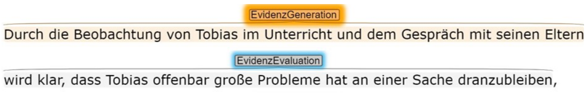
Figure 2: Annotation setup in MeD: Red indicates suggestions by a univ(ersal) or pers(onalised) model. The dashed boxes indicate annotations of texts that were already annotated in S1 or O1.

to A3 (randomly chosen among the five annotators) and instruct them to only accept epistemic activities if these coincide with what they would have annotated without suggestions and else manually annotate the correct text spans. We study the effectiveness of the suggestions (O1), the intra-annotator consistency (O2), the annotation bias induced by suggestions (O3), and the effectiveness of a personalised suggestion model (O4). Annotators A4 and A5 act as a control group, never receiving suggestions. We use an analogous setup for TeD except that there is no annotator A3.

To create gold standard annotations, we use majority voting and annotator meetings as Schulz et al. (2018), and we publish our final corpora.

4.1 Implementation

Annotation Tool Since we work with the same expert annotators as Schulz et al. (2018), we choose to also use the same annotation platform, INCEpTION (Klie et al., 2018), so that the expert annotators are already familiar with the interface. INCEpTION furthermore provides a rich API to integrate our suggestion models. As shown in Figure 3, annotation suggestions are shown in grey, distinguishing them clearly from differently coloured manual annotations. Suggestions can be easily accepted or rejected by single or double clicking. Additionally, manual annotations can be created as usual.


EvidenzGeneration

Durch die Beobachtung von Tobias im Unterricht und dem Gespräch mit seinen Eltern

EvidenzEvaluation

wird klar, dass Tobias offenbar große Probleme hat an einer Sache dranzubleiben,

Figure 3: Annotation suggestion (grey) and accepted suggestion (orange) in the INCEpTION platform.

Suggestion Models To suggest annotations, we use a state-of-the-art BiLSTM network with a conditional random field output layer (Reimers and Gurevych, 2017), which has proven to be a suitable architecture for related tasks (Ajjour et al., 2017; Eger et al., 2017; Levy et al., 2018). We train this model using the gold standard of Schulz et al. (2018), consisting of annotations for all texts from phases S1 and S2. The learning task is framed as standard sequence labelling with a BIO-encoding (Begin, Inside, Outside of a sequence) for the four epistemic activities hypothesis generation (HG), evidence generation (EG), evidence evaluation (EE), and drawing conclusions (DC). More precisely, each token is assigned one of the labels $(\{B, I\} \times \{HG, EG, EE, DC\}) \cup \{O\}$, where $B\text{-}HG$ denotes the first token of a HG segment, $I\text{-}HG$ denotes a continuation token of a HG segment (similarly for EG, EE, and DC), and O denotes a token that is not part of any epistemic activity.[3] We use this suggestion model in O1–O3.1 and call it *universal* (univ), as it learns labels obtained from all annotators of a domain.

For annotation phase O4.1, we train a *personalised* (pers) suggestion model for each annotator A1–A3, based on the epistemic activities identified by the respective annotator in phases S1 and S2. A personalised model thus provides suggestions tailored to a specific annotator. The idea of personalised models is that they may enable each annotator to accept more suggestions than possible with the universal model, which may lead to a speed-up in annotation time. Note, however, that each of these personalised models is trained using only 250 texts, 150 annotated by the respective annotator in S1 and 100 in S2. Instead, the universal model is trained using 650 (MeD) or 550 (TeD) texts.

We train ten models with different seeds for each setup (universal and three personalised for MeD and TeD), applying the same parameters for all of them: one hidden layer of 100 units, variational dropout rates for input and hidden layer of 0.25, and the *nadam* optimiser (Dozat, 2016). We furthermore use the German *fastText* word embeddings (Grave et al., 2018) to represent the input. We apply early stopping after five epochs without improvement. For the actual suggestions in our experiments, we choose the model with the best performance among the ten for each setup.

[3] We utilise the non-overlapping gold annotations of Schulz et al. (2018), where a preference order over epistemic activities was applied to avoid overlapping segments.

4.2 Suggestion Quality

Epistemic activity identification is a particularly hard discourse-level sequence labelling task, both for expert annotators and machine learning models. Before beginning with our annotation experiments, we evaluate our different suggestion models, as shown in Table 1. All models exhibit mid-range prediction capabilities, which we consider sufficient for automatic annotation suggestions. This is supported by Greinacher and Horn (2018), who find that suggestion models with an accuracy of at least 50 % improve annotation performance and speed for named entity recognition. Still, the overall performance for our task is clearly lower than in low-level tasks such as part-of-speech tagging, for which suggestions have been studied.

Domain	Test Data	univ	pers1	pers2	pers3
MeD	gold data	0.63	0.51	0.58	0.55
	ann. data	—	0.51	0.60	0.58
TeD	gold data	0.55	0.54	0.48	—
	ann. data	—	0.60	0.49	—

Table 1: Macro-F1 scores of the univ and pers models used in our experiments, evaluated on the gold and respective annotator-specific (ann.) annotations.

We evaluate the performance of the personalised models using both the annotations by the respective annotator and the gold annotations. The overall lower performance on the gold data shows that the personalised models indeed learn to predict the annotation style of the respective annotator. We also observe lower performance of the personalised models compared to the universal models, which can be attributed to the smaller amount of annotated texts used for training.

4.3 Evaluation and Findings

In this section, we examine the effects of annotation suggestions in detail, considering inter-annotator agreement, intra-annotator consistency, annotation bias and speed, as well as usefulness of suggestions and the impact of universal versus personalised suggestion models.

Effectiveness of Suggestions Since the annotation of epistemic activities involves determining spans as well as labels, we measure the inter-annotator agreement (IAA) in terms of Krippendorff's α_U (Krippendorff, 1995) as implemented in DKPro Agreement (Meyer et al., 2014). To evaluate the effects of suggestions on the annotations of our experts, we compare the IAA between annotators *with* suggestions (A1–A3) – henceforth called the SUGGESTION group – against the IAA between annotators *without* suggestions (A4–A5) – denoted as the STANDARD group. Table 2 details the IAA of the two groups across all annotation phases described in Figure 2.

	MeD			TeD		
Phase	ST	SU	SU/ST	ST	SU	SU/ST
S1	0.65	0.67	0.67	0.65	0.64	0.65
O1	0.71	0.73	0.70	0.73	0.77	0.73
O2	0.66	0.69	0.64	0.66	0.76	0.67
O3.1	0.60	0.60	0.59	0.73	0.80	0.71
O3.2	0.57	0.62	0.65	0.64	0.66	0.65
O4.1	-0.47	0.43	0.21	0.67	0.72	0.65
O4.2	0.60	0.68	0.60	0.67	0.74	0.71
O1–O4	0.48	0.64	0.59	0.69	0.75	0.68

Table 2: Inter-annotator agreement in terms of Krippendorff's α_U for ST(ANDARD) and SU(GGESTION) and their inter-group agreement (SU/ST). Bold: Phases in which models were used for SU.

First, we compare the overall IAA of both groups for the previous annotation phase S1 by Schulz et al. (2018) and all of our annotation phases O1–O4.2. We observe for TeD that the IAA of the SUGGESTION group is consistently higher than of the STANDARD group, as soon as annotators receive suggestions (starting in O1). Since the IAAs of the two groups were similar in S1, when no suggestions were given, we deduce that suggestions cause less annotation discrepancies between annotators in TeD. Below, we will investigate if this also introduces an annotation bias. For MeD, results are less clear, since the SUGGESTION group achieves only slightly higher IAA scores in most phases. Notable is the extreme outlier of the STANDARD group in O4.1. This is due to one annotator, whose EE (evidence evaluation) annotations deviated substantially from the other annotators. Considering the average IAA of our experiments without O4.1, we obtain very similar scores for the STANDARD (0.63) and SUGGESTION (0.66) group. Thus, there is little difference to reference phase S1, where SUGGESTION already yielded a 0.02 higher IAA. However, below we discuss the helpfulness and time saving of suggestions even in MeD.

Intra-Annotator Consistency In O2, we mixed 100 new texts with 50 texts the annotators saw previously during S1 or O1, but we did not inform the annotators about this setup. Table 3 shows the annotation consistency of each annotator in terms of *intra*-annotator agreement computed on those 50 double-annotated texts. Even a single annotator shows annotation discrepancies instead of perfect consistency, evidencing the difficulty of annotating epistemic activities. Since the intra-annotator agreement for annotators with suggestions (A1–A3) is similar to that without (A4–A5), we conclude that suggestions do not considerably change annotators' annotation decisions.

	Suggestion			Standard		
	A1	A2	A3	A4	A5	av.
MeD	0.74	0.76	0.79	0.78	0.80	0.77
TeD	0.77	0.64	—	0.72	0.70	0.71

Table 3: Intra-annotator agreement (in terms of Krippendorff's α_U) on double-annotated texts.

Annotation Bias The higher IAA in the Suggestion compared to the Standard group in TeD may indicate an annotation bias, i.e. a tendency that the Suggestion group prefers the predicted labels over the actual epistemic activities. We test this unwanted effect by comparing the human–machine agreement between the experts' annotations and the models' predictions (in terms of Krippendorff's α_U) for both annotators with and without suggestions. Table 4 shows that, in both MeD and TeD, annotators who receive suggestions, i.e. Suggestion in O1–O3.1 and in O4.1, consistently have a slightly higher agreement of about 0.1 than annotators without suggestions in these phases. This indicates an annotation bias due to suggestions. In MeD, this bias is preserved even if annotators do not receive suggestions anymore (Suggestion in O3.2 and O4.2), whereas in TeD the bias fades.

To further examine the gravity of the annotation bias, we compute the inter-group agreement, i.e. the average pairwise IAA between annotators *with* and *without* suggestions, denoted Su/St in Table 2. We find that this agreement is similar to the agreement within the Standard group for both MeD and TeD. In other words, an annotator with and an annotator without suggestions have the same level of agreement as two annotators without suggestions.

As a next step, we analyse the differences in the label distributions of the predictions and the Suggestion and Standard annotations. In MeD, the Suggestion annotators use EE (evidence evaluation) labels slightly more often, which can also be observed for the predictions. In TeD, the Suggestion annotators use fewer EE labels, but more HG (hypothesis generation) labels than Standard annotators, which again matches the tendency of the predicted labels. This effect is, however, very small, since all label distributions are close to each other. The Jensen-Shannon divergence (JSD) between the label distributions of the two annotator groups is consistently below 0.02 in all suggestion phases (O1–O3.1, O4.1) with an average JSD of 0.011 (MeD) and 0.009 (TeD). There is almost no difference to the JSD of the remaining phases (0.009 for MeD, 0.010 for TeD), indicating that the difference between the groups cannot be attributed to the suggestions.

		MeD			TeD		
		Su	St	diff.	Su	St	diff.
univ	S1	0.65	0.67	–0.02	0.55	0.52	+0.03
	O1	0.64	0.56	+0.08	0.52	0.42	+0.10
	O2	0.55	0.48	+0.07	0.50	0.42	+0.08
	O3.1	0.69	0.55	+0.14	0.54	0.40	+0.14
	O3.2	0.52	0.45	+0.07	0.51	0.49	+0.02
	O4.1	0.46	0.33	+0.13	0.47	0.39	+0.08
	O4.2	0.53	0.49	+0.04	0.40	0.40	+0.00
pers	**O4.1**	0.42	0.30	+0.12	0.49	0.41	+0.08
	O4.2	0.41	0.45	–0.04	0.34	0.32	+0.02

Table 4: Average α_U of annotators (in Su(ggestion) and St(andard)) with predictions of the univ and their pers model and diff(erence) between the groups. Bold: Phases in which models were used for Su.

We also compute the JSD of the Suggestion group and the predictions as well as the JSD of the Standard group and the predictions and find an average difference of the JSDs of -0.009 for MeD and < 0.001 for TeD, which indicates a small bias towards the suggested labels for MeD, but no obvious bias for TeD.

We finally analyse the disagreement within both groups of annotators. Figure 4 shows the distribution of the disagreements for TeD's Suggestion (left) and Standard group (right). We note that most disagreement occurs for EE labels. This is not surprising, as EE is the most frequently occurring label. The Suggestion group has a slightly higher disagreement for the DC (drawing conclusions) and HG labels, but overall, we do not observe substantial changes in the disagreement distribution, as also the disagreement for phases with-

Su	EG	EE	DC	HG
EG	-	4%	0%	3%
EE	4%	-	19%	15%
DC	0%	19%	-	9%
HG	3%	15%	9%	-

St	EG	EE	DC	HG
EG	-	4%	1%	1%
EE	4%	-	21%	18%
DC	1%	21%	-	5%
HG	1%	18%	5%	-

Figure 4: Disagreement among TeD annotators of the Su(ggestion) and St(andard) groups in phases with suggestions models (O1–O3.1 and O4.1).

out suggestions is up to 3 percentage points different between the two groups. For MeD, we find even smaller differences between the two groups.

Based on all analyses, we consider the annotation bias negligible, since suggestions do not cause negative annotation discrepancies compared to the standard annotation setup without suggestions.

Annotation Time Table 5 shows that nearly all annotators performed annotations faster in our experiments compared to previous annotations by Schulz et al. (2018), which can be attributed to the annotation experience they collected. We note that annotators in the Suggestion group (A1–A3) always speed up compared to previous annotations, whereas some of the annotators in the Standard group (A4–A5) slow down. Furthermore, on average, annotators in the Suggestion group exhibit a higher speed-up of annotation time: A1–A3 have a speed-up of 35 % compared to only 21 % for A4–A5 in MeD, and 20 % compared to only 11 % in TeD. Thus, suggestions make the annotation of epistemic activities more efficient.

		Suggestion			Standard	
	Phase	A1	A2	A3	A4	A5
MeD	S1–S2	1.92	2.13	1.82	3.78	1.94
	O1–O4	0.88	1.60	1.29	2.46	2.05
	speed-up	54 %	25 %	29 %	36 %	−6 %
TeD	S1–S2	2.73	2.91	—	2.57	2.31
	O1–O4	1.81	2.70	—	2.76	1.59
	speed-up	34 %	7 %	—	−8 %	31 %

Table 5: Average annotation time per text (in minutes) and speed-up of our compared to previous annotations.

Usefulness of Suggestions In addition to positive informal feedback from the Suggestion annotators about the usefulness of suggestions, we also perform an objective evaluation. As a new metric of usefulness, we propose the *acceptance rate* of suggestions. Table 6 shows that on average 56 % of the suggestions are accepted by the expert annotators in MeD and 54 % in TeD. Closer analysis reveals that in the many rejected cases, only the segment boundaries of suggestions were incorrect. This leads us to conclude that suggestions ease the difficult task of annotating epistemic activities.

	O1	O2	O3.1	O4.1	av.
MeD	58 %	49 %	62 %	54 %	56 %
TeD	59 %	55 %	60 %	43 %	54 %

Table 6: Percentage of accepted suggestions.

Personalised versus Universal Both in MeD and TeD, Table 2 shows a lower IAA in the Suggestion group when suggestions are given by a personalised model (O4.1) compared to the universal model (O1–O3.1). This can be explained by the fact that annotators are biased (see Table 4, O4.1 pers) towards different annotations due to suggestions by different personalised models.

We observe that annotators also accept fewer suggestions from the personalised than from the universal models (see Table 6), which can be attributed to the worse prediction performance of the personalised models (see Table 1). We conclude that our universal models exhibit more positive effects than the personalised models, as our goal is to create a gold standard corpus.

Discussion Our annotation study shows that annotation suggestions have various positive effects on the annotation of epistemic activities, despite the mediocre performance of our suggestion models. In particular, the agreement between annotators in TeD is increased without inducing a noteworthy annotation bias, and annotation time decreases in both MeD and TeD. Since the task of epistemic activity identification is a particularly hard one, both for humans and for machine learning models, we expect that the positive effects of annotation suggestions generalise to other discourse-level sequence labelling tasks.

5 Training Suggestion Models

The previous section established that annotation suggestions have positive effects on annotating epistemic activities. However, these suggestions were only possible since Schulz et al. (2018) had already annotated 550 reasoning texts in TeD and 650 in MeD, which were used to train our suggestion models. Envisioning suggestions for similar tasks with fewer or even no existing annotations, this section simulates suggestions of our universal models in this scenario. We experiment with different methods of training our models with only a small number of 'already annotated' texts and then continuously adjusting the models when 'newly annotated' texts become available.

5.1 Approach

We use the gold annotations of Schulz et al. (2018) for our experiments. The ongoing annotation of texts and the continuously increasing amount of available training data can be simulated as a (random) sequence S of texts t_i becoming available at each time step i, i.e. $S = t_1, t_2, \ldots, t_n$.

In addition to model adjustments at every time step, representing an online learning setup, we experiment with adjusting our models using *bundles* of texts (called batches by Read et al. (2012)). The models are thus only adjusted after each jth time step, where j is the bundle size. We experiment with bundle sizes 10, 20, 30, 40, and 50 and represent the single-step setup as bundle size 1.

The easiest way to adjust a suggestion model for each new bundle is to train a new model from scratch using the union of the new and all previously available bundles. We call this adjustment method RETRAIN and use bundle size 50. As a more advanced method, we suggest *repeatedly training* the existing model every time a new bundle of texts becomes available, i.e. the weights of the model are updated with each new bundle. We contrast two strategies for updating the model: the cumulative method (CUM) uses the union of the new and all previously available bundles of texts for training, whereas the incremental method (INC) uses only the new bundle.

For all model adjustment experiments, we use the architecture of our suggestion models described in Section 4.1. We report the average performance over ten runs for each setup (adjustment method, bundle size, domain). Our text sequence S has length 270. All models in the CUM and INC setup are initially trained on 10 texts before the repeated training with particular bundle sizes.

5.2 Results and Evaluation

We observe similar trends for MeD and TeD and therefore only present our MeD results in detail.

Model Performance Figure 5 shows the macro-F1 scores for the different adjustment methods with various bundle sizes. Using CUM, performance is very similar for all bundle sizes (1–50), thus represented by a single line in Figure 5. INC with bundle sizes 20 and 40 are omitted from the figure for readability. We observe that repeatedly training the model with CUM yields the same performance as RETRAIN, i.e. as training a new model from scratch for every new bundle. Furthermore, the performance of CUM rapidly increases with each bundle for the first 70 texts, reaching 0.5 macro-F1. The performance increase is more gradual thereafter, reaching 0.6 after 270 texts.

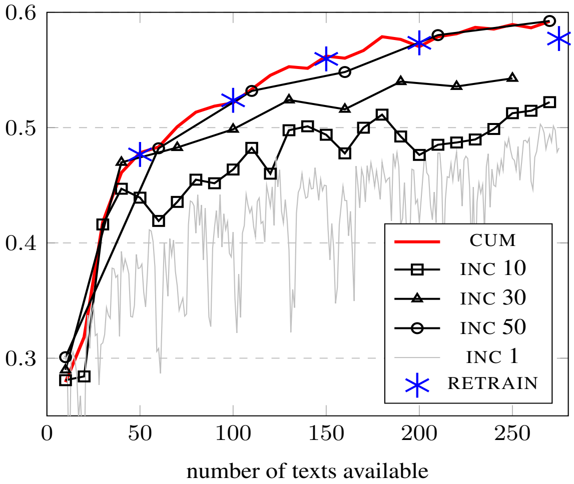


Figure 5: Macro-F1 after each adjustment using different methods and bundle sizes in MeD.

Using INC for repeated training, bundle size influences performance: A small bundle size of 1 to 20 results in unsteady performance, which increases in the long-run but shows decreases after training on some of the bundles. In contrast, bundle sizes of 30 and higher show a steady increase in performance, similar to CUM. However, after having trained on at least 70 texts, INC adjustments with a bundle size smaller than 50 yield lower performance results than CUM adjustments.

We conclude that to provide annotation suggestions, repeatedly training a model using INC with a bundle size of 30 or more can be a suitable alternative to CUM as well as to training models from scratch whenever new annotations become available, since the performance sacrifice is small.

Training Time Having observed only slight differences in the model performance using our different adjustment methods, Figure 6 reveals a clear distinction regarding the time needed to adjust to each new bundle (trends of bundle sizes not illustrated lie between those in the figure). While the training time using CUM *increases* with each new

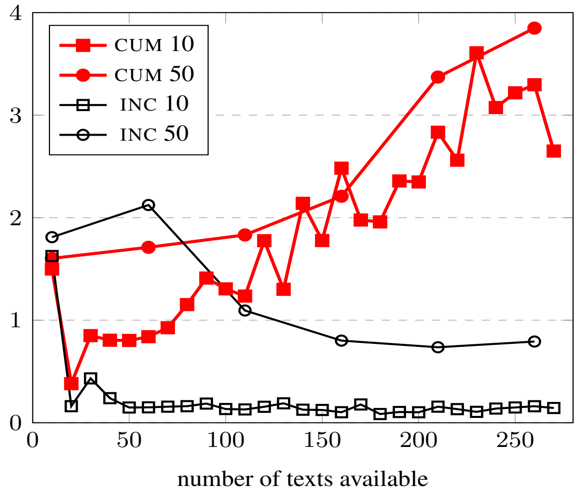


Figure 6: Training time (in minutes) for each adjustment using different methods and bundle sizes in MeD.

bundle, since each successive adjustment is performed with more data, the training time of INC *decreases* with each bundle, until reaching a stable minimum ranging from 8 seconds for bundle size 10 to 47 seconds for bundle size 50. This decrease in training time, despite the stable amount of data used for each adjustment, is due to a decrease in the number of epochs required for training and indicates that the texts used in previous training steps are beneficial for training the model on a completely new bundle of texts.

The RETRAIN method, not illustrated in the figure, requires far more time for adjustment than the repeated training methods. Training (from scratch) for 50 texts already takes 4.5 minutes, i.e. more than the CUM adjustment with 270 texts, and training for 270 texts takes 7.5 minutes.

Discussion Our results show that INC adjustments are the most time-efficient, with each adjustment being two to five times faster than CUM adjustments. In fact, in the CUM online learning setup (bundle size 1), the model adjustment time is similar to, and after 100 documents higher than the time needed for annotation (1–2 minutes per text as shown in Table 5). However, the adjustment times of CUM with a bundle size of 10 or higher and of RETRAIN, are lower than the time needed for annotating the respective bundle of texts. Thus, CUM training with bundles larger than 1 is feasible for continuously adjusting suggestion models in our annotation task (while only a small amount of data is available), despite the long training time compared to INC. Since CUM achieves the same performance results as RETRAIN but needs far less time for adjustment, we dismiss RETRAIN as a suitable method for training suggestion models.

6 Conclusion

We presented the first study of annotation suggestions for discourse-level sequence labelling requiring expert annotators, using the hard task of epistemic activity identification as an example. Our results show that even mediocre suggestion models have a positive effect in terms of agreement between annotators and annotation speed, while annotation biases are negligible.

Based on our experiments on training suggestion models, we propose for future annotation studies that annotation suggestions can be given after having annotated only a small amount of data (in our case 70 texts), which ensures a sufficient model performance (0.5 macro-F1). Since the exact number of texts required to reach sufficient model performance depends on the task, we suggest using continuous model adjustments from the start, ensuring flexibility as to when to start giving suggestions (namely whenever sufficient performance is achieved). If computational resources are an important factor, we propose the usage of INC training with a bundle size of 30 or higher to optimise performance and training time. If model performance is more important, we recommend CUM training using a small bundle size of 10 or 20 to improve suggestions in short intervals.

In our model adjustment experiments, we used gold annotations. To create them on the fly, annotation aggregation methods for sequence labelling (Simpson and Gurevych, 2018) can be used.

We expect our work to have a large impact on future work requiring expert annotations, in particular regarding new tasks with no or little available data, for example for legal (Nazarenko et al., 2018), chemical (Guo et al., 2014), or psychiatric (Mieskes and Stiegelmayr, 2018) text processing.

Acknowledgements

This work was supported by the German Federal Ministry of Education and Research (BMBF) under the reference 16DHL1040 (FAMULUS). We thank our annotators M. Achtner, S. Eichler, V. Jung, H. Mißbach, K. Nederstigt, P. Schäffner, R. Schönberger, and H. Werl. We also acknowledge Samaun Ibna Faiz for his contributions to the model adjustment experiments.

References

Luis von Ahn. 2006. Games with a Purpose. *Computer*, 39(6):92–94.

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. 2017. Unit segmentation of argumentative texts. In *Proceedings of the 4th Workshop on Argument Mining (ArgMin)*, pages 118–128, Copenhagen, Denmark.

Mariela Andrade, Eduardo Gasca, and Eréndira Rendón. 2017. Implementation of incremental learning in artificial neural networks. In *Proceedings of the 3rd Global Conference on Artificial Intelligence (GCAI)*, volume 50 of *EPiC Series in Computing*, pages 221–232.

Kalina Bontcheva, Hamish Cunningham, Ian Roberts, Angus Roberts, Valentin Tablan, Niraj Aswani, and Genevieve Gorrell. 2013. GATE Teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4):1007–1029.

Francisco M. Castro, Manuel J. Marin-Jimenez, Nicolas Guil, Cordelia Schmid, and Karteek Alahari. 2018. End-to-end incremental learning. In *Proceedings of the 15th European Conference on Computer Vision (ECCV)*, pages 241–257, Munich, Germany.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

Timothy Dozat. 2016. Incorporating nesterov momentum into adam. In *ICLR 2016 Workshop Track*, San Juan, Puerto Rico.

Hanne Martine Eckhoff and Aleksandrs Berdicevskis. 2016. Automatic parsing as an efficient pre-annotation tool for historical texts. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 62–70, Osaka, Japan.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural End-to-End Learning for Computational Argumentation Mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 11–22, Vancouver, Canada.

Paul Felt, Eric K. Ringger, Kevin Seppi, Kristian S. Heal, Robbie A. Haertel, and Deryle Lonsdale. 2014. Evaluating machine-assisted annotation in under-resourced settings. *Language Resources and Evaluation*, 48(4):561–599.

Frank Fischer, Ingo Kollar, Stefan Ufer, Beate Sodian, Heinrich Hussmann, Reinhard Pekrun, Birgit Neuhaus, Birgit Dorner, Sabine Pankofer, Martin R. Fischer, Jan-Willem Strijbos, Moritz Heene, and Julia Eberle. 2014. Scientific Reasoning and Argumentation: Advancing an Interdisciplinary Research Agenda in Education. *Frontline Learning Research*, 4:28–45.

Karën Fort and Benoît Sagot. 2010. Influence of Pre-Annotation on POS-Tagged Corpus Development. In *Proceedings of the 4th Linguistic Annotation Workshop (LAW)*, pages 56–63, Uppsala, Sweden.

Saurabh Garg, Tanmay Parekh, and Preethi Jyothi. 2018. Code-switched Language Models Using Dual RNNs and Same-Source Pretraining. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3078–3083, Brussels, Belgium.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*, pages 3483–3487, Miyazaki, Japan.

Robert Greinacher and Franziska Horn. 2018. The DALPHI annotation framework & how its pre-annotations can improve annotator efficiency. arXiv:1808.05558.

Yufan Guo, Diarmuid Ó Séaghdha, Ilona Silins, Lin Sun, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2014. Crab 2.0: A text mining tool for supporting literature review in chemical cancer risk assessment. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING): System Demonstrations*, pages 76–80, Dublin, Ireland.

Mahnoosh Kholghi, Laurianne Sitbon, Guido Zuccon, and Anthony Nguyen. 2017. Active learning reduces annotation time for clinical concept extraction. *International Journal of Medical Informatics*, 106:25–31.

Jan-Christoph Klie, 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 (COLING): System Demonstrations*, pages 5–9, Santa Fe, NM, USA.

Klaus Krippendorff. 1995. On the Reliability of Unitizing Continuous Data. *Sociological Methodology*, 25:47–76.

Ran Levy, Ben Bogin, Shai Gretz, Ranit Aharonov, and Noam Slonim. 2018. Towards an argumentative content search engine using weak supervision. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 2066–2081, Santa Fe, NM, USA.

Todd Lingren, 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.

Marie-Jean Meurs, Caitlin Murphy, Nona Naderi, Ingo Morgenstern, Carolina Cantu, Shary Semarjit, Greg Butler, Justin Powlowski, Adrian Tsang, and René Witte. 2011. Towards evaluating the impact of semantic support for curating the fungus scientic literature. In *Proceedings of the 3rd Canadian Semantic Web Symposium (CSWS)*, pages 34–39, Vancouver, Canada.

Christian M. Meyer, Margot Mieskes, Christian Stab, and Iryna Gurevych. 2014. DKPro Agreement: An Open-Source Java Library for Measuring Inter-Rater Agreement. In *Proceedings of the 25th International Conference on Computational Linguistics: System Demonstrations (COLING)*, pages 105–109, Dublin, Ireland.

Margot Mieskes and Andreas Stiegelmayr. 2018. Preparing Data from Psychotherapy for Natural Language Processing. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*, pages 2896–2902, Miyazaki, Japan.

Adeline Nazarenko, François Levy, and Adam Wyner. 2018. An Annotation Language for Semantic Search of Legal Sources. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*, pages 1096–1100, Miyazaki, Japan.

Beatriz Pérez-Sánchez, Oscar Fontenla-Romero, and Bertha Guijarro-Berdiñas. 2010. An incremental learning method for neural networks in adaptive environments. In *Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Barcelona, Spain.

Martin Potthast, Tim Gollub, Kristof Komlossy, Sebastian Schuster, Matti Wiegmann, Erika Patricia Garces Fernandez, Matthias Hagen, and Benno Stein. 2018. Crowdsourcing a Large Corpus of Clickbait on Twitter. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 1498–1507, Santa Fe, NM, USA.

Jesse Read, Albert Bifet, Bernhard Pfahringer, and Geoff Holmes. 2012. Batch-incremental versus instance-incremental learning in dynamic and evolving data. In *Advances in Intelligent Data Analysis XI*, pages 313–323. Berlin/Heidelberg: Springer.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 338–348, Copenhagen, Denmark.

Sophie Rosset, Cyril Grouin, Thomas Lavergne, Mohamed Ben Jannet, Jérémy Leixa, Olivier Galibert, and Pierre Zweigenbaum. 2013. Automatic named entity pre-annotation for out-of-domain human annotation. In *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse*, pages 168–177, Sofia, Bulgaria.

Dan Roth. 2017. Incidental Supervision: Moving beyond Supervised Learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pages 4885–4890, San Francisco, CA, USA.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive Neural Networks. arXiv:1606.04671.

Claudia Schulz, Christian M. Meyer, and Iryna Gurevych. 2019. Challenges in the Automatic Analysis of Students' Diagnostic Reasoning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, Honolulu, HI, USA. (to appear).

Claudia Schulz, Christian M. Meyer, Michael Sailer, Jan Kiesewetter, Elisabeth Bauer, Frank Fischer, Martin R. Fischer, and Iryna Gurevych. 2018. Challenges in the Automatic Analysis of Students' Diagnostic Reasoning. arXiv:1811.10550.

Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building End-to-end Dialogue Systems Using Generative Hierarchical Neural Network Models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, pages 3776–3783, Phoenix, AZ, USA.

Toru Shimizu, Nobuyuki Shimizu, and Hayato Kobayashi. 2018. Pretraining Sentiment Classifiers with Unlabeled Dialog Data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 764–770, Melbourne, Australia.

Edwin Simpson and Iryna Gurevych. 2018. Bayesian Ensembles of Crowds and Deep Learners for Sequence Tagging. arXiv:1811.00780.

Maria Skeppstedt, Carita Paradis, and Andreas Kerren. 2016. PAL, a tool for Preannotation and Active Learning. *Journal for Language Technology and Computational Linguistics*, 31(1):81–100.

Brett R. South, Danielle Mowery, Ying Suo, Jianwei Leng, scar Ferrndez, Stephane M. Meystre, and Wendy W. Chapman. 2014. Evaluating the effects of machine pre-annotation and an interactive annotation interface on manual de-identification of clinical text. *Journal of Biomedical Informatics: Special Issue on Informatics Methods in Medical Privacy*, 50:162–172.

Morgan Ulinski, Julia Hirschberg, and Owen Rambow. 2016. Incrementally learning a dependency parser to support language documentation in field linguistics. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pages 440–449, Osaka, Japan.

Rob Voigt, Nicholas P. Camp, Vinodkumar Prabhakaran, William L. Hamilton, Rebecca C. Hetey, Camilla M. Griffiths, David Jurgens, Dan Jurafsky, and Jennifer L. Eberhardt. 2017. Language from police body camera footage shows racial disparities in officer respect. *Proceedings of the National Academy of Sciences*, 114(25):6521–6526.

Seid Muhie Yimam, Chris Biemann, Richard Eckart de Castilho, and Iryna Gurevych. 2014. Automatic annotation suggestions and custom annotation layers in webanno. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, pages 91–96, Baltimore, MD, USA.