Interactive Evidence Detection: train state-of-the-art model out-of-domain or simple model interactively?

Chris Stahlhut
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Research Training Group KRITIS
Department of Computer Science, Technische Universitt Darmstadt
https://www.ukp.tu-darmstadt.de/

Abstract

Finding evidence is of vital importance in research as well as fact checking and an evidence detection method would be useful in speeding up this process. However, when addressing a new topic there is no training data and there are two approaches to get started. One could use large amounts of out-of-domain data to train a state-of-the-art method, or to use the small data that a person creates while working on the topic. In this paper, we address this problem in two steps. First, by simulating users who read source documents and label sentences they can use as evidence, thereby creating small amounts of training data for an interactively trained evidence detection model; and second, by comparing such an interactively trained model against a pre-trained model that has been trained on large out-of-domain data. We found that an interactively trained model not only often out-performs a state-of-the-art model but also requires significantly lower amounts of computational resources. Therefore, especially when computational resources are scarce, e.g. no GPU available, training a smaller model on the fly is preferable to training a well generalising but resource hungry out-of-domain model.

1 Introduction

Evidence is a crucial prerequisite for research, forming an opinion, and fact checking. Scholars spend vast amounts of time reading through countless books and other documents to find evidence relevant to their research; fact checkers read through innumerable documents to find evidence to (in)validate popular claims.

Evidence Detection (ED) aims at supporting these activities by finding textual evidence and thereby reducing the amount of reading required by a human. In this paper, we define evidence similar to Shnarch et al. (2018) as a sentence that either supports or contradicts a controversial topic, e.g. *we should ban gambling* and is categorisable as *expert opinion*, *anecdote*, or *study data* (figure 1). This is similar to premise detection in argument mining, but requires the additional filtering for these particular types.

> *A 2010 Australian hospital study found that 17% of suicidal patients admitted to the Alfred Hospital's emergency department were problem gamblers.*

Figure 1: An example piece of evidence.

In this paper, we focus on the following scenario. Suppose a group of fact checkers is evaluating a set of claims that are gaining popularity. They start by distributing the claims among each other and downloading relevant articles from Wikipedia. They then intend to use an ED method to help them collect the evidence but are faced with the question of where to get the training data from. First, they could use the data that has been compiled for previous claims; or second, train a model interactively. The former approach introduces a domain shift, while the latter turns ED into a small data problem.

From this we developed our research questions

(1) Does a simple but interactively trained model out-perform a state-of-the-art model that was trained on out-of-domain data?

(2) What amount of in-domain training data is required to out-perform the state-of-the-art model trained on out-of-domain data?

We investigated the first research question by comparing the results of static models that have been trained on out-of-domain data with ones that learn on the in-domain data. As out-of-domain model we chose BERT (Devlin et al., 2018) because it performs well on both ED and Argument Mining (AM) (Reimers et al., 2019). As in-domain trained model we chose a topic agnostic

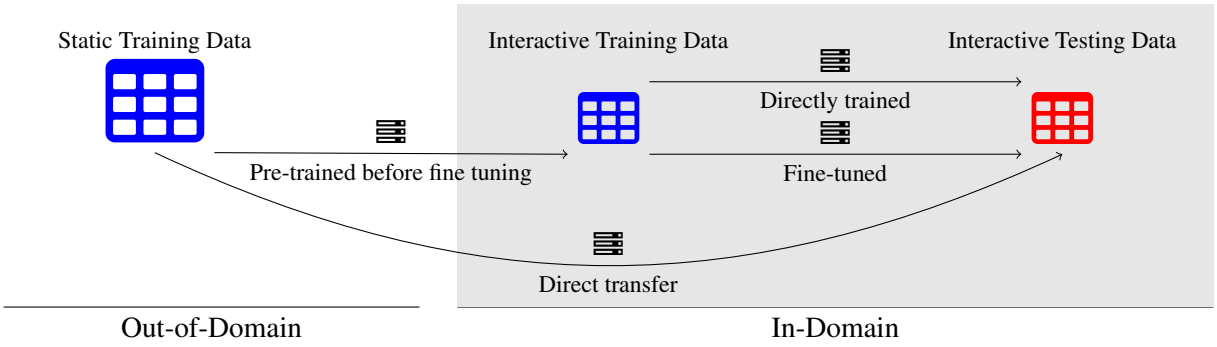


Figure 2: The relation between the out-of-domain and in-domain datasets and different training setups.

BiLSTM which performed well in in-domain experiments in AM (Stab et al., 2018). We chose a topic agnostic model because each user is working on only one topic which doesn't change between samples and therefore contains no additional information. To address the cold-start problem, we also evaluated a similar topic agnostic model that has been pre-trained on the out-of-domain data and was then fine-tuned on the in-domain data. We did not fine-tune BERT on the in-domain data, because we consider the datasets too small. Figure 2 shows the relationship between the different domains and models. To investigate our second research question, we used simulated users who each trained a personalised ED model interactively. We then compared the quality of the interactively trained or fine-tuned models with the static BERT trained on out-of-domain data. We also investigated the robustness of our results interactively fine-tuning a model for AM. We chose AM, because it is similar in that it contains arguments (pro and contra) on a controversial topic, such as *nuclear energy*.

The contributions of this paper are three fold. (1) A much simpler model can out-perform a state-of-the-art model when given in-domain training data, (2) that often only a few documents for training are required, and (3) a more realistic evaluation interactive ED than random downsampling and the datasets used in our experiments.

2 Related Work

This paper touches three areas of research, namely the overarching field of claim validation, the task domain (ED and AM) with small data, and the interaction of Natural Language Processing (NLP) components with users.

Claim Validation Reasoning about the validity of a particular claim can be separated into three sub-tasks: document retrieval to find documents related to the claim, ED to find the relevant pieces of evidence that support or contradict the claim, and Textual Entailment (TE) to determine whether the claim follows from the evidence. The FEVER shared tasks follows this approach (Thorne et al., 2018; Thorne and Vlachos, 2019). Other approaches, such as TwoWingOS (Yin and Roth, 2018) and DeClarE (Popat et al., 2018) combine the ED and TE models into a single end-to-end method. Ma et al. (2019) used two pre-trained models, one for ED and one for TE which are then jointly fine-tuned. While presenting promising results, all of these approaches rely on static models that are trained beforehand and do not learn from the user.

Evidence detection and argument mining Much focus of ED has been in on supporting decision making (Hua and Wang, 2017) or to find evidence for debating (Rinott et al., 2015; Aharoni et al., 2014). Evidence detection can be seen as a sub-task of AM. Argument mining is an established task within NLP with different foci, e.g. parsing arguments from student essays (Stab and Gurevych, 2017) or extracting topic related argumentative sentences from Wikipedia articles (Levy et al., 2018). Still, the cold-start problem for new domains and topics remains and multiple approaches have been suggested to address it. One approach is to increase the generalisability of a learned AM model, either by adding topic information (Stab et al., 2018) or by using distant supervision with automatically extracted data from debate portals (Al-Khatib et al., 2016). A similar method was used by Shnarch et al. (2018) who

combined weakly and strongly labeled data to reduce the necessary amount of expensive to create strongly labeled data for ED. Schulz et al. (2018a) on the other hand, used multi-task learning with artificially shrunk target datasets. However, artificially shrinking a dataset to a pre-defined number of samples is not a realistic simulation method for interactive learning because it does not take the content of a document and resulting bias in the training data into account. While the previous approaches mostly worked with large amounts of data, some work with smaller datasets was conducted in the medical domain. For instance finding and classifying evidence in the abstracts of research articles (Shardlow et al., 2018; Mayer et al., 2018). However, neither of these approaches consider learning interactively from users.

Interactive NLP Combining NLP components with direct human interactions generally serves either the system or the user. Focussing on the system side is generally done to support the process of annotation for a dataset, such as improving dependency parsing of historical documents (Eckhoff and Berdicevskis, 2016) via pre-annotation. Moreover, learning directly from users is beneficial from the first sentence on in dependency parsing (Ulinski et al., 2016). Another common approach is to use active learning to reduce the amount of data to train a model (Kasai et al., 2019; Lin et al., 2019). While these approaches are beneficial in creating annotated data or speeding up the training of a model, they focus on the goal of the system. Focussing on the goal of the users, on the other hand, is all about benefiting the user, for instance supporting teachers in evaluating the diagnostic reasoning abilities of students (Schulz et al., 2018b). The INCEpTION (Klie et al., 2018) platform also focusses on the user's goals by learning from users to assist them in their annotation work. However, all these approaches assume the task to be independent from the individual user, which Stahlhut et al. (2018) showed to not be the case for ED. This is especially important, because the system's recommendations do influence what the user annotates (Fort and Sagot, 2010). The SHERLOCK system (P.V.S. et al., 2018) does does offer user specific results, but is not focussed on ED but multi-document summarisation.

3 Interactive Evidence Detection

For our experiments, we defined ED as extracting sentences from a collection of documents D that are evidential[1] regarding a controversial topic. Interactive ED considers the same task in combination with a user who provides the documents and order in which they are processed, as well as corrections of the predictions of the ED model m.

3.1 User simulation

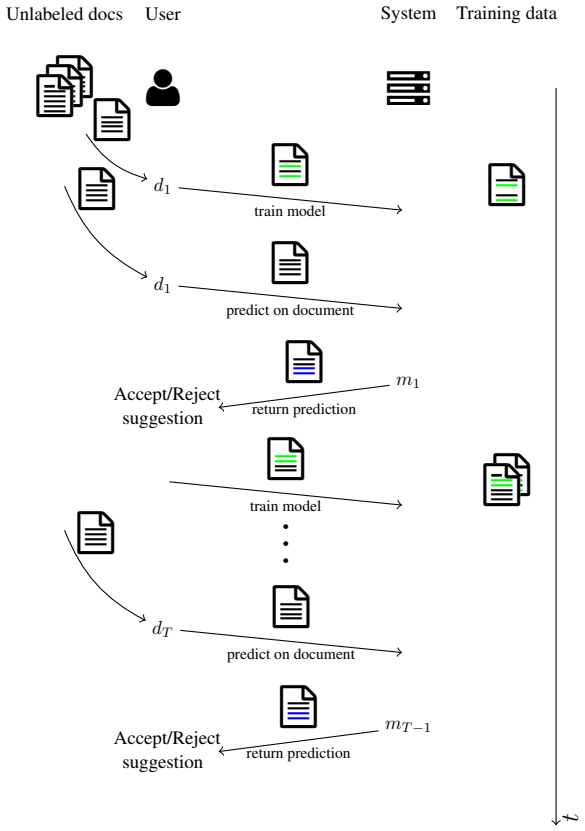


Figure 3: The user picks one unlabeled document and annotates the evidential sentences. After processing the document, it gets added to the training data for a newly trained model. Afterwards, the user picks the next document which contains suggestions from the model.

Each user sorts all documents that are relevant to their topic in alphabetical order and proceeds to read one document at a time. While reading the first document $d_1 \in D$, the user labels each sentence they find evidential regarding the topic as evidence. After reading the entire document d_1 they proceed to the next one d_2 without returning to the

[1]For simplicity, we are referring to arguments also as evidence.

previous one d_1. The document d_1 is then added to the training data and the interactive training begins with the training of the model m_1.

When the user opens the next document d_2, it already contains suggestions regarding evidential sentences by the model m_1. The user then accepts correct suggestions of evidential sentences, rejects incorrect suggestions, and labels missed pieces of evidence. After the user finishes reading and correcting the labels of all sentences, the amount of training data again increases and a new model m_2 is trained on them. This cycle continues until the user opens the last document d_T, $T = |D|$ which shows suggestions made by the previously trained model m_{T-1}. Figure 3 illustrates our simulation and interactive training.

3.2 Measure of work-load

In our simulation, we also required a measure that allows us to compare the amount of work a user has to perform to correct the suggestions of different models. This includes not only the incorrectly suggested evidential sentences, but also the missing ones. We therefore defined an error rate that accounts for incorrect, as well as missing suggestions of evidence. Formally, we defined the *error rate* E as the sum of the *false discovery rate* and *false omission rate*, or

$$E = (1 - P) + (1 - R), \tag{1}$$

with P being the precision and R being the recall on the evidence class.

3.3 Measurement of minimal amount of training data

To answer our second research question, we needed to measure the minimum amount of training documents μ required to out-perform a static model, which we defined as

$$\mu = \begin{cases} \min \{t \mid \forall i \in \{t, \ldots, |\mathcal{R}|\} : \mathcal{R}_i < \mathcal{B}\} & , \exists t : \mathcal{R}_t < \mathcal{B} \\ |\mathcal{R}| & , \text{otherwise}, \end{cases} \tag{2}$$

where $\mathcal{R}$ is the sequence of error-rates through time $t \in \{1, \ldots, T\}$ and $\mathcal{B}$ is the average error-rate of the baseline.

4 Data and Models

4.1 Datasets and data preparation

We used three different datasets as in-domain data for our evaluation. For ED we used two datasets, namely *ED-ACL-2014* published by Aharoni et al. (2014) and *ED-EMNLP-2015* published by Rinott et al. (2015). As out-of-domain data, we used a dataset published by Shnarch et al. (2018), named *ED-ACL-2018*.[2] For AM, we used the dataset provided by Stab et al. (2018).

Data preparation To run the user simulation, we needed to convert the data from collections of evidential sentences to documents with sentences labeled as evidential or not.[3] We converted all three datasets into topic related collections of documents with sentential annotations. That means we took all documents that are relevant to a particular topic and labeled all sentences that are evidential towards this topic in each of these documents. For evidential sequences that are more than one sentence long, we first segmented them into individual sentences via NLTK.[4] To avoid problems due to errors in the sentence segmentation, we ignored all evidential sentences with a length of less than three tokens. The resulting datasets are highly biased towards non-evidential sentences.

ED-ACL-2014 The first ED dataset contains 12 topics and 315 articles from Wikipedia as source with 143 containing evidence. The individual pieces of evidence can be up to 16 sentences long with about half being exactly one sentence and about 90% being up to three sentences in length.

ED-EMNLP-2015 The second dataset consists of 58 topics, 19 of which are for development purposes, and 2.3k hypotheses. Of these hypotheses, 1.4k are supported by at least one piece of evidence from Wikipedia articles. The dataset uses 1.3k Wikipedia articles as source for the evidence, of which 547 contain at least one piece of evidence. We decided to exclude twelve of the test topics due to their large overlap with the ED-ACL-2014 dataset, leaving 27 test topics.

ED-ACL-2018 As out-of-domain data for the pre-trained ED models we chose the dataset presented by Shnarch et al. (2018). It contains 4k topic evidence pairs as training data and 1.7k pairs as testing data. We pre-trained models exclusively on the training data so that we could use the testing data for comparison with published literature.

[2] The topics and number of documents for each topic can found in the supplementary material

[3] The source code for the data preparation and experiments can be found under https://github.com/UKPLab/fever2019-interactive-evidence-detection.

[4] https://www.nltk.org/

Table 1 shows an overview of the statistics of all three ED datasets.

	Documents	Sentences	Evidence
ED-ACL-2014			
train test	143	20649	1318
ED-EMNLP-2015			
train dev	170	28540	2300
train test	234	35877	2646
ED-ACL-2018			
train	–	4065	1499
test	–	1718	683

Table 1: Statistics on the ED datasets.

Argument mining The AM corpus consists of about 25k sentences that are evidential (distinguishing supporting from contradicting evidence) or non-evidential regarding one of eight topics. The sentences of each topic were extracted from the 50 highest ranking documents retrieved by an external search engine. In our processing, we labeled the evidential sentences in the original documents. This lead to a change in number of sentences and pieces of evidence as table 2 shows. When separated into in- and out-of-domain data, we selected all documents of one topic as in-domain data, and the training data of the other seven as out-of-domain data.

	Documents	Sentences	Evidence
Original	–	25492	11139
Converted	400	39577	11538[5]

Table 2: Statistics on the AM dataset before and after the data preparation.

4.2 Models

We built two interactively trained models, $\text{bilstm}_{\text{direct}}$ and $\text{bilstm}_{\text{fine}}$, and used BERT as static model trained on the out-of-domain data. We refer to the $\text{bilstm}_{\text{fine}}$ after its pre-training but before additional fine tuning as $\text{bilstm}_{\text{pre}}$. Table 3 shows the models and which data they are trained on, ouf-of-domain, in-domain, or both. We decided to use a BiLSTM with 100 nodes, a dense layer for classification, and no input for the topic for these experiments because the in-domain training data is small and always specific to a single topic. All interactively trained models used 100-dimensional GloVe embeddings (Pennington et al., 2014) as input features and a dropout of 0.5 after the embedding layer and before the classification layer. We addressed the class imbalance by weighting the classes similar to King and Zeng (2001) using the implementation provided by scikit-learn.[6] To reduce the effect of the random initialisation, we repeated all experiments with 10 different randomisation seeds.

	Training domain	
	Out-of-Domain	In-Domain
$\text{bilstm}_{\text{direct}}$	no	yes
$\text{bilstm}_{\text{pre}}$	yes	no
$\text{bilstm}_{\text{fine}}$	yes	yes
BERT	yes	no

Table 3: Model label depending on the training data.

$\text{bilstm}_{\text{direct}}$ The directly trained model was trained as described above and received no additional input. We trained this model for 10 epochs in each iteration with one additional training document.

$\text{bilstm}_{\text{pre}}$ The pre-trained model uses the same architecture than the directly trained one. We changed no hyper-parameter except the number of epochs compared to the directly trained model. That means, we trained the $\text{bilstm}_{\text{pre}}$ model for five epochs on the out-of-domain training data and used a learning rate of 0.001 with a dropout of 0.5.

$\text{bilstm}_{\text{fine}}$ For fine-tuning, we replaced the classification layer of the $\text{bilstm}_{\text{pre}}$ model with a new one and trained this new layer for five epochs with a learning rate of 0.001. Afterwards, we unfroze the other layers and trained the complete network for five more epochs with a learning rate of 0.001. This is similar to gradual unfreezing, presented by Howard and Ruder (2018).

BERT Short for Bidirectional Encoder Representations from Transformers. We chose the BERT base model (Devlin et al., 2018) as static model, since it outperforms previously published models on both tasks (Reimers et al., 2019). We fine-tuned it for three epochs on the out-of-domain data. We provided the model with the candidate sentence, as well as the topic, because we fine-tuned the model across multiple topics of the training data and used the same model for prediction

[5] The number varies due to duplicated evidential sentences. There are 11128 unique pieces of evidence in the converted dataset.

[6] https://scikit-learn.org/

	Macro values across both classes			Evidence only		
	F1	Precision	Recall	F1	Precision	Recall
ED-ACL-2014						
$\text{bilstm}_{\text{direct}}$	0.509 (0.033)	0.514 (0.028)	0.526 (0.039)	0.117 (0.058)	0.091 (0.055)	0.183 (0.053)
$\text{bilstm}_{\text{fine}}$	0.481 (0.043)	0.518 (0.018)	**0.553 (0.047)**	**0.139 (0.064)**	0.088 (0.045)	**0.373 (0.118)**
BERT	**0.540 (0.052)**	**0.590 (0.055)**	0.538 (0.048)	0.118 (0.098)	**0.238 (0.105)**	0.094 (0.096)
ED-EMNLP-2015						
$\text{bilstm}_{\text{direct}}$	**0.572 (0.062)**	0.566 (0.050)	0.613 (0.075)	**0.225 (0.133)**	0.176 (0.114)	0.340 (0.160)
$\text{bilstm}_{\text{fine}}$	0.544 (0.063)	0.553 (0.046)	**0.631 (0.089)**	0.212 (0.132)	0.145 (0.101)	**0.453 (0.212)**
BERT	0.550 (0.060)	**0.596 (0.084)**	0.558 (0.081)	0.143 (0.118)	**0.251 (0.169)**	0.143 (0.171)
Argument Mining						
$\text{bilstm}_{\text{fine}}$	0.681 (0.021)	0.698 (0.014)	0.739 (0.021)	0.620 (0.027)	0.490 (0.034)	**0.848 (0.015)**
BERT	**0.754 (0.016)**	**0.747 (0.015)**	**0.779 (0.015)**	**0.676 (0.023)**	**0.599 (0.033)**	0.780 (0.038)

Table 4: The results are macro-averaged across all topics with the standard deviations shown in parenthesis.

across all topics in the in-domain data. We used a PyTorch based implementation provided by Huggingface[7].

5 Experiments

5.1 Evaluation of pre-trained models

We evaluated the pre-trained models on the testing data of their pre-training domain. That means that in the case of ED, we trained and evaluated the models on the ED-ACL-2018 dataset. For AM, we conducted a leave-one-topic-out evaluation, training on the training data of the training topics and evaluated on the testing data of the left-out topic.

	F1	Precision	Recall	Accuracy
ED-ACL-2018				
$\text{bilstm}_{\text{pre}}$	0.609	0.620	0.608	0.639
BERT	0.781	0.809	0.770	0.802
Argument Mining				
$\text{bilstm}_{\text{pre}}$	0.624	0.647	0.632	–
BERT	0.795	0.800	0.800	–

Table 5: Results of the pre-trained models on their respective training domain test data. The results are macro-averaged for F1, Precision, and Recall.

The table 5 shows the quality of the pre-trained models for both the ED and AM experiments with macro-averaged F1, precision, and recall. BERT clearly out-performed the topic agnostic model $\text{bilstm}_{\text{pre}}$ by a margin of almost 18pp macro F1 score for ED. For AM, BERT also clearly out-performed the topic agnostic model by about 17pp macro-F1 score in binary classification of evidence/no-evidence.

5.2 Static evaluation

In the static evaluation, we compared the performance of the static model with the interactively trained ones after having been trained with all training documents. We conducted the experiments in a leave-one-document-out fashion for each topic separately. Table 4 shows the results of the static evaluation. We found that although BERT reached the highest macro F1 score on the ED-ACL-2014 dataset, it did not perform better than the fine-tuned model when looking at the evidence F1 score due to its lower recall. On the ED-EMNLP-2015 dataset, all three models improved compared to the ED-ACL-2014 dataset. Furthermore, both interactively trained models improved more than BERT, increasing the gap when performing better.

We conducted the experiments on the AM data also in a leave-one-document-out fashion for each interactively processed topic, using the training data of the other topics for pre-training. We found that BERT out-performed $\text{bilstm}_{\text{fine}}$ by about 7pp macro F1 score, which is a considerable smaller margin than before fine-tuning. Moreover, the difference varies between the individual metrics, being closer in evidence F1 score and in evidence recall the $\text{bilstm}_{\text{fine}}$ model even out-performs BERT.

5.3 Interactive evaluation

To avoid irregularities due to changes in number of pieces of evidence and length of a document between different amounts of training data, we cal-

[7] https://github.com/huggingface/pytorch-pretrained-BERT

culated the error-rate in a leave-one-document out fashion. This means, instead of calculating the error-rate, defined by (1), on the next document the user opens which might have a different number of pieces of evidence, we calculated it on a left-out one. The left-out document then remains the same across the experiment with increasing number of training documents. We then repeated this process with each document being left-out once. As before, we repeated the experiments with ten different randomisation seeds.

Table 6 shows that the $\text{bilstm}_{\text{fine}}$ model reached a lower error-rate and therefore requires less work for the user to correct than BERT on the ED-ACL-2014 dataset. It already did so after few training documents.

Id	Docs	$\text{bilstm}_{\text{fine}}$		BERT
		μ	E	E
0	6	6.000	1.761	**1.230**
1	19	3.800	**1.526**	1.677
2	10	10.000	1.752	**1.617**
3	11	1.000	**1.288**	1.742
4	13	6.300	**1.655**	1.737
5	10	5.200	**1.612**	1.772
6	13	1.000	**1.535**	1.898
7	6	1.300	**1.665**	1.830
8	13	5.500	**1.525**	1.681
9	15	10.200	**1.307**	1.426
10	20	6.200	**1.397**	1.560
11	7	1.000	**1.445**	1.846

Table 6: Number of documents and minimum number of training documents μ to reach a smaller error-rate E than BERT for the $\text{bilstm}_{\text{fine}}$ model for each topic on the ED-ACL-2014 dataset. The values for μ and E are averaged across all left-out documents and repeated experiments.

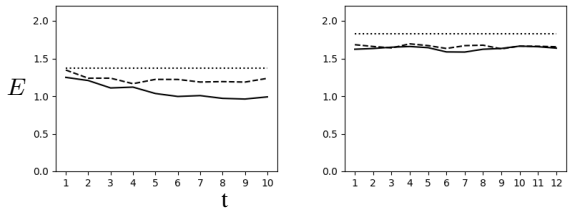
On the ED-EMNLP-2015 dataset (table 7), we found that both interactively trained models generally out-perform the static BERT and that they reach a lower error-rate often already after one or two training documents. When comparing the interactively trained models, we found that the $\text{bilstm}_{\text{fine}}$ often reaches slightly better results than the $\text{bilstm}_{\text{direct}}$ model. BERT reached the lowest overall error-rate on topic 6 which contained only two documents. We selected the topics 1, 5, 18 and 8 for a more detailed analysis with a focus on the amount of work a user would have to do to correct the suggestions of a model. Figure 4 shows that for topic 1 (figure 4a the $\text{bilstm}_{\text{fine}}$ model outperformed the $\text{bilstm}_{\text{direct}}$ model. In the case of the topics 18 and 5 (figures 4b and 4c), we found that the both interactively trained model learned at a similar rate. For topic 8 (figure 4d), on the other hand, neither interactively trained model reached the performance of BERT.

Id	Docs	$\text{bilstm}_{\text{direct}}$		$\text{bilstm}_{\text{fine}}$		BERT
		μ	E	μ	E	E
0	5	5.000	1.739	5.000	1.837	**1.586**
1	11	1.000	1.194	1.000	**0.932**	1.373
2	4	4.000	1.932	4.000	1.981	**1.226**
3	4	1.000	**1.432**	1.200	1.474	1.793
4	3	1.000	**1.235**	1.000	1.247	1.799
5	13	1.000	1.643	1.000	**1.591**	1.829
6	2	2.000	2.000	2.000	2.000	**0.723**
7	14	1.000	1.123	1.000	**1.011**	1.472
8	4	4.000	1.805	4.000	1.592	**1.289**
9	4	1.000	1.244	1.000	**1.182**	1.607
10	17	17.000	1.592	5.000	**1.268**	1.385
11	8	1.400	**1.307**	1.000	1.329	1.636
12	15	14.000	1.554	4.800	**1.433**	1.543
13	9	8.800	1.486	6.500	**1.405**	1.416
14	12	2.200	1.484	1.200	**1.297**	1.683
15	12	3.100	1.500	1.000	**1.366**	1.643
16	14	1.000	1.213	1.000	**1.049**	1.724
17	3	2.300	1.449	1.700	**1.401**	1.490
18	25	1.000	1.190	1.000	**1.152**	1.517
19	5	4.700	1.960	5.000	2.000	**1.903**
20	4	2.600	1.862	2.200	**1.688**	1.949
21	10	1.000	1.039	1.000	**1.019**	1.606
22	12	1.100	1.240	1.000	**1.147**	1.431
23	6	3.000	1.585	3.000	**1.478**	1.990
24	6	1.000	1.456	1.000	**1.350**	1.965
25	7	2.700	1.590	3.800	**1.461**	1.923
26	5	1.000	1.217	1.000	**1.151**	1.871

Table 7: Number of documents and minimum number of training documents μ to reach a smaller error-rate E then BERT for the $\text{bilstm}_{\text{direct}}$ and $\text{bilstm}_{\text{fine}}$ models for each topic on the ED-EMNLP-2015 dataset. The values for μ and E are averaged across all left-out documents and repeated experiments.

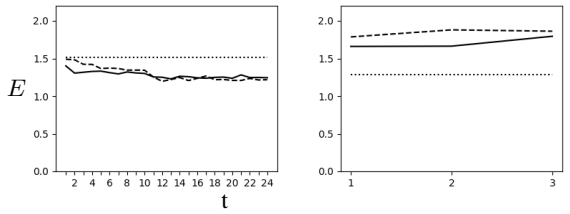
6 Discussion

To understand the difference in quality between the ED-ACL-2014 and ED-ACL-2015 dataset we hypothesise that the annotators gained more experience which lead to a more consistent evidence annotation. This might also beneficial for machine learning. When creating the ED-ACL-2014 dataset, Aharoni et al. (2014) stated that they used five annotators that searched Wikipedia independently from each other for evidence on the same topic. Afterwards, they used five different annotators to accept or reject these annotations. Rinott et al. (2015) used the same process, although not for twelve but 58 topics. This means that the same annotator had the opportunity to work on many more topics than when constructing the ED-ACL-2014 dataset.



(a) Topic 1

(b) Topic 5



(c) Topic 18

(d) Topic 8

Figure 4: The average error-rates E of the $\text{bilstm}_{\text{fine}}$ (solid lines), $\text{bilstm}_{\text{direct}}$ (dashed lines), and BERT (dotted lines) through time t.

To evaluate this hypothesis, we chose the twelve topics from the ED-EMNLP-2015 dataset which we excluded due to their large overlap with the topics of the ED-ACL-2014 dataset. While being very similar in topic and using the same Wikipedia articles as sources, they are not identical. Some hypotheses were added and others were removed. The $\text{bilstm}_{\text{fine}}$ and $\text{bilstm}_{\text{direct}}$ models improved considerable ($\approx$9pp and $\approx$13pp respectively in evidence F1 score) on the later created dataset compared to the previously created one. To our surprise however, we found that in stark contrast to the other models, BERT's performance decreased. We assume that due to the larger number of topics, the annotators gained more experience and created more consistent annotations, making the ED-EMNLP-2015 dataset more machine learning friendly.

	F1	Precision	Recall
$\text{bilstm}_{\text{direct}}$	0.243 [+0.126]	0.195 [+0.104]	0.345 [+0.162]
$\text{bilstm}_{\text{fine}}$	0.233 [+0.094]	0.155 [+0.067]	0.509 [+0.136]
BERT	0.110 [−0.008]	0.290 [+0.052]	0.073 [−0.021]

Table 8: The results of the evidence class scores and are macro-averaged across the previously held out topics of the ED-EMNLP-2015 dataset. The values in the brackets are the difference the to ED-ACL-2014 dataset.

While BERT performed very well on the ED-ACL-2018 dataset, when tested on the ED-ACL-2014 and ED-EMNLP-2015 datasets, its performance dropped significantly. We developed two hypotheses that might explain this drop.

The topic labels used in the ED-EMNLP-2015 dataset are worded as debate motions which is different from the wording in the ED-ACL-2018 dataset. In the latter dataset, the topics are worded directly as a controversial statement, e.g. *We should ban gambling*, which is different from the wording as a debate motion *This house would ban gambling*. To test this hypothesis, we selected three topics from the ED-EMNLP-2015 dataset which also appear in ED training domain for BERT. We then updated the topic label to be the same as the one used in the training data for BERT and evaluated the effect this had on the performance. We found that the modification of the topic label to be more like the one used while training BERT increased the evidence F1 score by 1pp (table 9); the wording of the topic label therefore cannot be the reason for the low performance of BERT.

	F1	Precision	Recall
in-domain topic label	0.077	0.213	0.050
out-of-domain topic label	0.087	0.262	0.060

Table 9: The results show only the evidence class and are macro-averaged across the three selected topics.

In our second hypothesis, we suggest that the sentence segmentation into partial evidence caused the dramatic drop in recall between the ED-ACL-2018 and other ED datasets. If so, then using the complete pieces of evidence that consist of multiple sentences would be classified correctly with much higher probability. We therefore also evaluated the recall that BERT reached on the multiple sentence long pieces of evidence on the previously selected three topics. We found that not segmenting the evidence increased the performance by almost 4pp to 0.098. This is too small to explain the observed drop in performance.

A possible influence on the minimum number of training documents μ is also the order in which the documents are processed. The error-rate of topic 1 in the ED-EMNLP-2015 dataset first decreased with the first four training documents and then varied. For topic 8, the error-rate increased with the amount of increasing training data. It is therefore possible that can also be dependent on the order of documents. However, as we defined the minimum number of training documents μ as the first document after which it out-performs the

baseline, which means that there will be no subsequent reduction in performance below the baseline, we think that the influence is small and can be treated as additional noise. We decided to use an alphabetical order, because it is deterministic and does not add additional degrees of freedom which an ranking based order would, e.g. by using term frequency versus TF-IDF.

7 Conclusion

In this paper we investigated the question of whether to use large amounts of out-of-domain data or small amounts of interactively generated data to train an ED or AM model. To answer this question, we simulated users who read documents relevant to a particular topic and while doing so, generated training data for the interactively trained models. We also converted three existing datasets, two ED and one AM dataset, into collections of topic relevant documents of labeled sentences. We then used the simulated users working on the newly created corpora to interactively train a model and compared it to a state-of-the-art static model, in our case BERT, that was fine-tuned on the out-of-domain data. We found that especially for ED the interactively trained models out-performed BERT in evidence F1 score. We also found that it would take the user less work to correct the predictions of an interactively trained model. Moreover, it often does so after only a few iterations. In AM, we found that although BERT performed best, it does so by a small margin.

We conclude from these results that unless computational resources are abundant, e.g. a GPU is available for training as well as prediction, it is better to train a model interactively, even if it is no longer state-of-the-art. This is especially important when considering constraints placed on interactive system that are used by multiple users in parallel. In the future, we intend apply these results to support real users in finding evidence by interactively training an ED model.

Acknowledgements

This work has been supported by the German Research Foundation (DFG) as part of the Research Training Group KRITIS No. GRK 2222/1. Calculations for this research were conducted on the Lichtenberg high performance computer of the TU Darmstadt.

References

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A Benchmark Dataset for Automatic Detection of Claims and Evidence in the Context of Controversial Topics. In *Proceedings of the First Workshop on Argumentation Mining*, pages 64–68. Association for Computational Linguistics.

Khalid Al-Khatib, Henning Wachsmuth, Matthias Hagen, Jonas Köhler, and Benno Stein. 2016. Cross-Domain Mining of Argumentative Text through Distant Supervision. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1395–1404, San Diego, California. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*.

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. The COLING 2016 Organizing Committee.

Karën Fort and Benoît Sagot. 2010. Influence of Pre-Annotation on POS-Tagged Corpus Development. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 56–63, Uppsala, Sweden. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. *arXiv:1801.06146 [cs, stat]*.

Xinyu Hua and Lu Wang. 2017. Understanding and Detecting Supporting Arguments of Diverse Types. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 203–208, Vancouver, Canada. Association for Computational Linguistics.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource Deep Entity Resolution with Transfer and Active Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5851–5861, Florence, Italy. Association for Computational Linguistics.

Gary King and Langche Zeng. 2001. Logistic Regression in Rare Events Data. *Political Analysis*, 9(2):137–163.

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: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

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*, pages 2066–2081, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Bill Yuchen Lin, Dong-Ho Lee, Frank F. Xu, Ouyu Lan, and Xiang Ren. 2019. AlpacaTag: An Active Learning-based Crowd Annotation Framework for Sequence Tagging. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 58–63, Florence, Italy. Association for Computational Linguistics.

Jing Ma, Wei Gao, Shafiq Joty, and Kam-Fai Wong. 2019. Sentence-Level Evidence Embedding for Claim Verification with Hierarchical Attention Networks. page 12.

Tobias Mayer, Elena Cabrio, and Serena Villata. 2018. Evidence Type Classification in Randomized Controlled Trials. In *Proceedings of the 5th Workshop on Argument Mining*, pages 29–34, Brussels, Belgium. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Brussels, Belgium. Association for Computational Linguistics.

Avinesh P.V.S., Benjamin Hättasch, Orkan Özyurt, Carsten Binnig, and Christian M. Meyer. 2018. Sherlock: A system for interactive summarization of large text collections. *Proceedings of the VLDB Endowment*, 11(12):1902–1905.

Nils Reimers, Benjamin Schiller, Tillman Beck, Johannes Daxenberger, and Iryna Gurevych. 2019. Classification and Clustering of Arguments with Contextualized Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page to appear, Florence, Italy. Association for Computational Linguistics.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show Me Your Evidence – an Automatic Method for Context Dependent Evidence Detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 440–450, Lisbon, Portugal. Association for Computational Linguistics.

Claudia Schulz, Steffen Eger, Johannes Daxenberger, Tobias Kahse, and Iryna Gurevych. 2018a. Multi-Task Learning for Argumentation Mining in Low-Resource Settings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 35–41, New Orleans, Louisiana. Association for Computational Linguistics.

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

Matthew Shardlow, Riza Batista-Navarro, Paul Thompson, Raheel Nawaz, John McNaught, and Sophia Ananiadou. 2018. Identification of research hypotheses and new knowledge from scientific literature. *BMC Medical Informatics and Decision Making*, 18(1):46.

Eyal Shnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it Blend? Blending Weak and Strong Labeled Data in a Neural Network for Argumentation Mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–605, Melbourne, Australia. Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2017. Parsing Argumentation Structures in Persuasive Essays. *Computational Linguistics*, 43(3):619–659.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic Argument Mining from Heterogeneous Sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674, Brussels, Belgium. Association for Computational Linguistics.

Chris Stahlhut, Christian Stab, and Iryna Gurevych. 2018. Pilot Experiments of Hypothesis Validation Through Evidence Detection for Historians. In *Proceedings of the First Biennial Conference on Design of Experimental Search & Information Retrieval Systems*, volume 2167 of *CEUR Workshop Proceedings*, pages 83–89, Bertinoro, Italy.

James Thorne and Andreas Vlachos. 2019. Adversarial attacks against Fact Extraction and VERification. *arXiv:1903.05543 [cs]*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Morgan Ulinski, Julia Hirschberg, and Owen Rambow. 2016. Incrementally Learning a Dependency Parser to Support Language Documentation in Field Linguistics. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 440–449, Osaka, Japan. The COLING 2016 Organizing Committee.

Wenpeng Yin and Dan Roth. 2018. TwoWingOS: A Two-Wing Optimization Strategy for Evidential Claim Verification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 105–114, Brussels, Belgium. Association for Computational Linguistics.