Domain Adaption with Active Learning

Andreas Svenningsen
aksv@itu.dk

Ida Wennergaard
idwe@itu.dk

Joke Heinks
bhei@itu.dk

Sebastian Wærling
swae@itu.dk

Abstract

Annotating textual data for NER requires substantial human resources to obtain high quality annotations applicable for NER tagger models. Both active learning (AL) and transfer learning seeks to cope with data scarcity. Based on earlier studies it is not evident to what extent AL can optimize the training process of BERT. In this paper we investigate if AL can help improve the performance with small amounts of data in a domain adaptation setting where BERT has initially been pre-fine-tuned on a source domain and hereafter fine-tuned on a target domain using AL. We compare uncertainty and diversity based query strategies with a random query strategy. Based on our results it is not evident, whether the uncertainty and diversity based query strategies outperform random selection of samples.

1 Introduction

Named Entity Recognition (NER) is a subtask of natural language processing, which seeks to extract entities and classify them into predefined entity types Xu et al. (2023). NER has shown to be a challenging task due to large variations in entity names and flexibility in entity mentions. Most existing NER models rely on massive amounts of annotated data, making it hard to directly apply them to data-limited domains Xu et al. (2023).

Transfer-learning and active learning both seeks to overcome this challenge. The goal of transfer learning is to use the information, which is present in a source dataset and leverage it to improve the performance of the model on a target dataset. Fine-tuning is the process of adapting a pre-trained model for a specific target domain. Fine-tuning deep neural networks is a popular approach for domain adaptation. However this approach requires a significant amount of labeled data from the target domain to be successful.Ma et al. (2019) Active learning (AL) addresses the issue of data scarcity by focusing on information rich sentences reducing the data that needs to be annotated, thereby reducing the cost of labelling data.

We propose to use active learning to overcome the problem of limited NER data in the target domain, with the goal of improving domain adaption in the NER setting. We want to quantify how the amount of data used to transfer learning correlates with performance. Specifically we want to measure the amount of data which can be used to achieve similar results when using all available data. We use a newly annotated Danish dataset DANSK Enevoldsen et al. (2024) for simulating domain adaption. Our GitHub repository contains the necessary code and instructions on how to reproduce our results[1].

The research question for this project is; *How can we best adapt a baseline model to a new domain by using active learning?*

2 Related Work

The aim of active learning is to reduce the cost of labeling data by querying informative samples. Uncertainty-based AL have gained popularity, and several query strategies have been proposed to estimate the sample uncertainty. Mosqueira-Rey et al. (2023) lists least confidence (LC), margin of confidence, ratio of confidence, and entropy. To the best of our knowledge least confidence is the most used of these methods.

The traditional query strategies has their limitations. Shen et al. (2017) propose Maximum Normalized Log-Probability (MNLP) as a modification to LC that seeks to solve the issue that you risk introducing a bias towards longer sentences using LC.

Zhang and Plank (2021) and Karamcheti et al. (2021) points to another issue of the traditional query strategies. They divide the training samples into hard-to-learn, ambiguous and easy-to-learn. They show that AL strategies prefers to acquire outliers, which are hard or even impossible for the model to learn, thereby decreasing the performance.

Ein-Dor et al. (2020) showed that active learning boosted BERT performance in NLP classification tasks. However, the same study concluded that it was not obvious whether active learning could

[1] GitHub Repository

outperform the already excellent performance of fine tuning BERT on little data.

3 Methods

3.1 Baseline Model Architecture

BERT Devlin et al. (2019) (Bidirectional Encoder Representations from Transformers) is designed to learn bidirectional token representations by jointly conditioning on both left and right context in all layers. The pre-trained BERT can be fine-tuned and then used for a wide range of down-stream tasks, for example Named Entity Recognition (NER).

We used multilingual BERT base cased, which is pre-trained on Wikipedia in 104 different languages Hugging Face. BERT has a fixed input sequence of 512 tokens, meaning sentences shorter than 512 tokens are padded using padding tokens [PAD]. Each token has an input representation which is the sum of the corresponding token, segment, and position embeddings Devlin et al. (2019). An example input could look like this: [CLS], J, ##yl, ##land, Øst, ##jylland, TOP, [SEP], Julian, ##e, Marie, K, ##jer, ##bo, ##e, [PAD], ..., [PAD].

BERT can be fine-tuned by adding a NER head. The NER head is illustrated on Figure 1. It takes the final contextualized embedding from BERT and transforms them to NER tags. It consists of a linear layer and a softmax layer. We further added a dropout layer between the contextualized embeddings and the linear layer with a dropout probability of 0.1, the same value used in Enevoldsen et al. (2024). The dropout layer forces the model to learn alternative patterns in the data. For the optimizer, we used the adam optimizer using L2 weight decay of 0.01 as in the paper by Enevoldsen et al. (2024). The model architecture is summarized in Figure 1.

3.2 Transfer learning

The goal of transfer learning is to use the information, which is present in a source dataset and leverage it to improve the performance of the model on a target dataset. We apply transfer learning by training the parameters of our model on a source dataset, and using the same model to train on the target dataset for fine-tuning. Lee et al. (2018)

3.3 Active learning

The goal of active learning is to find an informativeness metric, that gives a subset of samples to train on. We apply pool based active learning, which means that the entire collection of data (or a subset) is evaluated and ranked in order to select the best element to annotate. There are three types of sampling strategies: random, uncertainty and diversity. For the uncertainty sampling we used two methods.

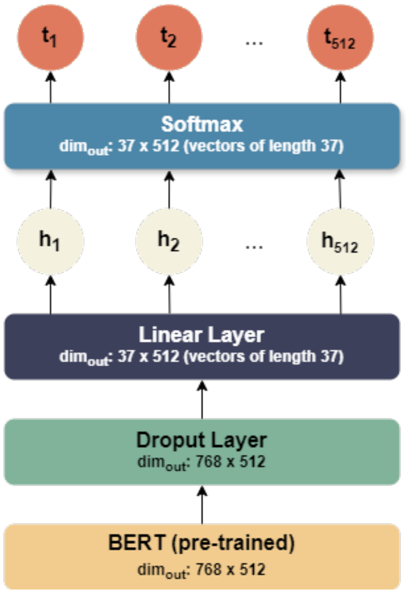


Figure 1: Pre-trained BERT with NER-head

- **Least confidence** chooses the token, where the most likely label has the lowest confidence.
- **Margin of confidence** chooses the token, where the difference in confidence between the top two labels is smallest.

For comparison we also use random sampling, which chooses tokens at random. Further, we used batch sampling, where multiple samples get labeled each round. (Mosqueira-Rey et al., 2023) A challenge, is that the uncertainty score is calculated on token level. We take an average of the token uncertainty scores to get the uncertainty score for the sentence. Summing or using the minimum tends to introduce an undesirable bias towards longer sentences (Luo et al., 2023)

3.4 Data Map

As explained in Related Work, the uncertainty based AL methods prefer to choose the samples that are hard for the model to learn.

Following the approach in Swayamdipta et al. (2020), we build a Dataset Map for the DANSK Dataset. The Dataset Map can be used to categorise training instances into: easy-to-learn, ambiguous and hard-to-learn, by plotting them along the axis:

- **Confidence** is the mean model probability for the gold label over the E epochs

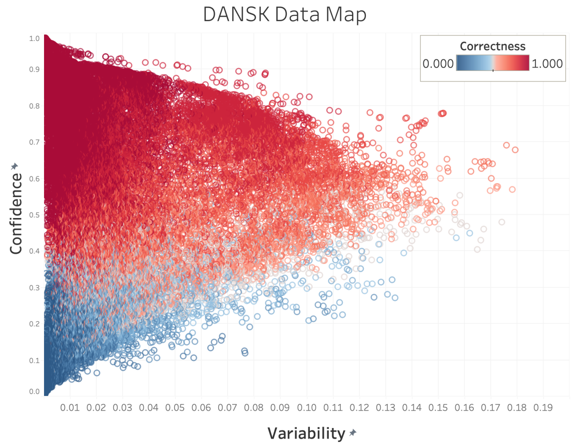


Figure 2: Data map DANSK train set (token-level). The x-axis shows **variability** and y-axis, the **confidence**; the colors indicate **correctness**. The top-left corner of the data map corresponds to **easy-to-learn** examples, the bottom left corner corresponds to **hard-to-learn** examples, and examples to the right are **ambiguous**

- **Variability** is the standard deviation of the confidence
- **Correctness** is calculated as the fraction of correct predictions across the E epochs

The statistics are calculated on token-level across the first 10 epochs, with a separate model for each of the 7 domains. "O" tokens are filtered out.

The hard-to-learn samples seen in dark blue are a problem, because the model does not improve on these samples during training. Hard-to-learn samples can for example be very unusual or ambiguous samples. An example is the sentence *"GUD BEVARE DANMARK"*. Which gets the worst average confidence score in the Conversation domain on 0.53 and the average maximum likelihood in the sentence is 0.59. The only entity tag in the sentence is the GPE "DANMARK", which should be relatively easy to learn because it is the name of the country, so it should occur many times. However, in the training data for Conversation it is only mentioned once in upper case. Because our model is case sensitive, this worsens the performance. Further, when the tokenizer doesn't recognize a word it will be split up more, which increases the weight in the average uncertainty measure. Both factors making the query strategies more likely to chose it for annotation.

3.5 Vocabulary

To avoid that problem, we also used queries based on diversity. For that we used vocabulary based queries, that are based on the BERT tokens, in the text. We used a vector with the frequencies of the Bert tokens in the source domain and training sentence. We then used the dot product of those two vectors as a similarity score. The queries then took the sentences with the lowest similarity score, to find data that shows the differences to the source domain. That also has the problem of being a very simple measure. And it assumes that the difference in performance can be attributed to the different tokens.

4 Experimental Setup

4.1 Data

We have chosen to work with the recently introduced dataset DANSK Enevoldsen et al. (2024), which is derived from the Danish gigaword corpus Derczynski et al. (2021). DANSK consist of 7 domains, *News, Conversation, Legal, Web, Wiki & Books, Social Media, dannet*, see appendix A. DANSK was introduced to solve the limitations of multi domain NER evaluation for Danish data. Furthermore they introduced fine-grained entity types, the same as Ontonotes 5.0, with 18 entity types. The entity types were not Beginning-Inside-Outside (BIO) tagged. Since this is the standard for NER tagging Jurafsky and Martin (2023), we chose to convert the original entity types to BIO tags, resulting in 36 tags. This was done with no information loss.

4.2 Evaluation Metrics

To evaluate the performance of our experiments we used the span F1-score. Span F1-score is computed by comparing predicted tag sequences with gold tag sequences for each sentence. We have made some examples in Table 1.

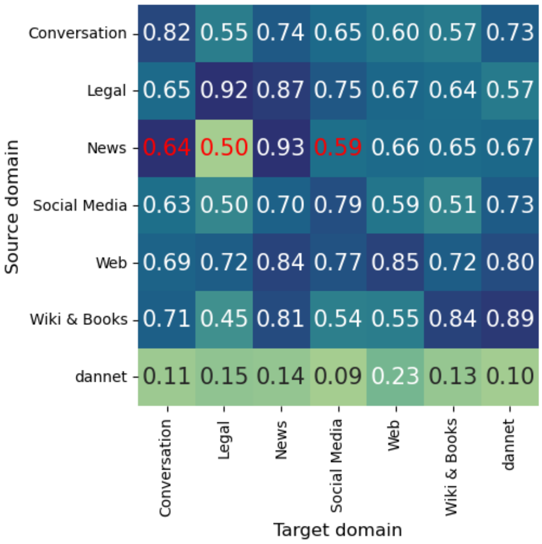
Type	**Jylland**	**Østjylland**	**TOP**	
Gold	B-GPE	B-GPE	O	
Pred	B-GPE	B-GPE	O	2TP
Pred	O	O	B-GPE	FP + 2FN
Pred	B-GPE	O	O	TP + FN
Pred	O	B-PER	O	FP + 2FN

Table 1: Gold sequence and different prediction sequences. One prediction sequence can get both False Positive and False Negatives. It can also incur a True Positive and a False prediction.

The span F1-score is defined as the harmonic mean between recall and precision. Recall measures how many entities in gold we found. Precision measures how many of our predicted entities are correct. For NER we want both high precision and recall,, why span F1 is a good evaluation metric.

4.3 Experimental Settings

We consider the baseline model as the model pre-fine-tuned on data from the chosen source domain and tested on the target domains without applying AL. With seven domains there are 49 possible combinations of source and target domain. Due to computational limitations we have chosen one source domain and three target domains. To select the suitable domains we have pre-fine-tuned BERT on each domain and tested it on all seven domains. The result can be seen in Figure 3.



Source domain \ Target domain	Conversation	Legal	News	Social Media	Web	Wiki & Books	dannet
Conversation	0.82	0.55	0.74	0.65	0.60	0.57	0.73
Legal	0.65	0.92	0.87	0.75	0.67	0.64	0.57
News	0.64	0.50	0.93	0.59	0.66	0.65	0.67
Social Media	0.63	0.50	0.70	0.79	0.59	0.51	0.73
Web	0.69	0.72	0.84	0.77	0.85	0.72	0.80
Wiki & Books	0.71	0.45	0.81	0.54	0.55	0.84	0.89
dannet	0.11	0.15	0.14	0.09	0.23	0.13	0.10

Figure 3: Span F1-scores for the baseline-model fine-tuned on one domain (source) and tested on another domain (target). The chosen domains are highlighted in red.

We chose news as the source domain. There are a couple of reasons for this. The model performs well when it is pre-fine-tuned and tested on data from the News domain. Furthermore, models pre-fine-tuned on other domains and tested on the News domain perform well, which indicates that the domain is fairly general. Finally, the performance of the model pre-fine-tuned on the news domain is lacking, indicating room for improvement. We chose Conversation, Legal and Social Media as our target domains. The target domains have a good size, meaning we have more data to simulate on. For each target domain we apply four AL strategies: random, least confidence, margin of confidence and vocabulary. To mitigate the challenge of the AL methods acquiring the hard-to-learn samples, we have chosen to set the pool size to be smaller than the entire dataset, and to train each model on a random subset of data of the target domain before applying the AL strategies. We set the pool size to be 20% of the dataset and the query-size to 5% of the dataset. The pool samples are chosen at random. The model is reset and retrained each time a new set of samples are labelled. This is repeated 10 times for each source and target domain. The reported results are the average.

5 Results and Analysis

5.1 Active learning

Figure 4 shows the performance of the models for different proportions of training data. The bottom grey line shows the performance of the pre-fine-tuned model, which is the same results marked with red in Figure 3. The top grey line shows the performance you get when pre-fine-tuning jointly on the source and target domain.

The general trend is that the span-F1 increases as the percentage of data used increases. For all three domains, the span-F1 increases only until about 40-50% of the dataset is labelled, and hereafter the curve flattens out. However, the results shown in Figure 4 indicate that our uncertainty and diversity sampling strategies achieve similar results as the random sampling strategy. By applying each of the four AL sampling strategies it is possible to surpass the pre-fine-tune performance for all domains, and reach the performance of jointly pre-fine-tuning on both source and target domain.

For Legal we see a relative large increase in span-F1 compared to the baseline performance even with a small amount of data. As we saw in Figure 3, the models pre-fine-tuned on other domains consistently had a lower performance on Legal, indiciating that the domain is less similar to the other domains. An explanation of the large increase could therefore be that the Legal domain is less similar to News than Conversation and Social Media, and therefore benefits more from training on a small subset of data. The model fine-tuned on Legal reaches a higher span-F1 score than the two other models. This could suggest, that the Legal domain is not hard to learn. Despite the relatively high span-F1 score the model does not reach the performance of the model, that has been jointy fine-tuned on News and Legal. This suggest that there is a difference in how much each domain benefits

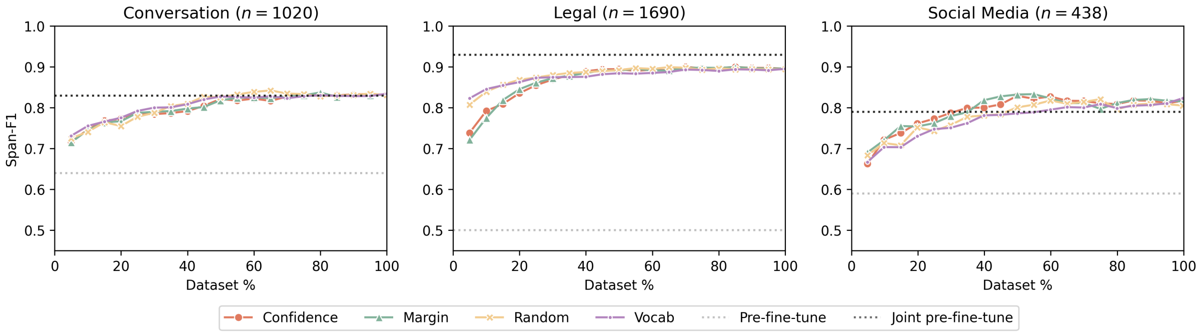


Figure 4: Span-F1 score for each target domain and AL method for different proportions of data used. The **pre-fine-tuned** model is fitted on news, and tested on the target domain. The **joint pre-fine-tuned** model is fitted on news and the target domain jointly. Note that the first query is based on random sampling for all query strategies, hence the starting point for each does not indicate how well the query strategy performs.

from the information in the source domain.

6 Discussion

The comparison to random sampling show that our query strategies do not give a substantial improvement in performance. For uncertainty based queries the reason could be a problem of prioritising hard-to-learn samples. This could be mitigated by including more randomness in the queries. However, this does not solve the problem but instead tries to reduce the harm. A better solution would be to create a metric which takes the problem of hard-to-learn samples into account for example with the variance. Diversity queries avoid this problem, because they do not focus on hard-to-learn samples. However, our vocabulary queries did not outperform random sampling, so there are other problems with our strategies. The problem could be, that we have not found a metric which explains the difference in performance between domains.

In each iteration of AL we incrementally query 5% of the target domain. If an annotator were to do this, the annotator would label the selected sentences, upload these tags, and the model would be fitted again. When adapting only one domain, this induces a lot of down time when the model is training, where the annotator is not labeling data. To optimize the annotation time, it is therefore good to have multiple domain adaption models running, otherwise the cost of having an annotator would be large compared to the results. Another option to optimize annotation time is to possibly not resetting the model after each AL iteration. Currently the model is reset after each iteration to avoid overfitting. A solution to not overfit could be to use the newly labeled samples, while sampling a fraction of already labeled samples. This would overcome the challenge of overfitting on early labeled samples. The best solution is to steer away from using uncertainty based AL methods, and instead use diversity based AL methods, such as the vocabulary strategy. These AL methods find and order the sentences to be labeled from the start. This means there is no waiting time for the annotator.

Our results do not show whether the pre-fine-tuning improves the performance in the active learning process. For that we would need to find a query strategy that effectively utilises the information from the source domain. And generally we do not know how the performance would look with the pretrained BERT without pre-finetuning.

7 Conclusion

Based on our findings it was possible to perform NER transfer learning from a source to a target domain by applying active learning. The performance when fine-tuning to the target domain increased for all three target domains and by applying active learning it was possible to obtain almost the same performance with approximately half of the amount of data. However, the performance when applying either the confidence or margin AL strategy was not able to surpass the performance of the random strategy for two out of three target domains. Thus it can be concluded that the AL strategies did not improve the performance in general.

Acknowledgements

We would like to thank Kenneth Enevoldson for making the DANSK dataset available at our disposal and Rob van der Goot for providing guidance during the project.

References

Leon Derczynski, Manuel R. Ciosici, Rebekah Baglini, Morten H. Christiansen, Jacob Aarup Dalsgaard, Riccardo Fusaroli, Peter Juel Henrichsen, Rasmus Hvingelby, Andreas Kirkedal, Alex Speed Kjeldsen, Claus Ladefoged, Finn Årup Nielsen, Jens Madsen, Malte Lau Petersen, Jonathan Hvithamar Rystrøm, and Daniel Varab. 2021. The Danish Gigaword Corpus. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics*. NEALT.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Kenneth Enevoldsen, Emil Trenckner Jessen, and Rebekah Baglini. 2024. Dansk and dacy 2.6.0: Domain generalization of danish named entity recognition.

Team Hugging Face. bert-base-multilingual-cased. Accessed on 05/13/2024.

Daniel Jurafsky and James H. Martin. 2023. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3rd edition. Prentice Hall PTR, USA.

Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher Manning. 2021. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7265–7281, Online. Association for Computational Linguistics.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2018. Transfer learning for named-entity recognition with neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Haocheng Luo, Wei Tan, Ngoc Nguyen, and Lan Du. 2023. Re-weighting tokens: A simple and effective active learning strategy for named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12725–12734, Singapore. Association for Computational Linguistics.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain adaptation with BERT-based domain classification and data selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 76–83, Hong Kong, China. Association for Computational Linguistics.

Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. 2023. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review*, 56(4):3005–3054.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Jingyun Xu, Changmeng Zheng, Yi Cai, and Tat-Seng Chua. 2023. Improving named entity recognition via bridge-based domain adaptation.

Mike Zhang and Barbara Plank. 2021. Cartography active learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 395–406, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Data

The distribution of sentences in the dataset.

Domain	**Train**	**Validation**	**Test**
Conversation	1020	122	130
Dannet	18	4	3
Legal	1690	234	239
News	346	36	39
Social Media	439	51	64
Web	6661	826	783
Wiki & Books	1361	166	182

Table 2: Amount of training, validation and test sentences in each domain

B Result

The span-F1 scores for our experiments can be seen in Table 3. The table shows which domains the model is trained on and whether there has been applied active learning or not. Finetuned means that BERT has been finetuned directly on the domain. Active learning means that AL has been used to adapt the model from one domain to another.

The first part of table 3 shows that the models perform much better when fine-tuned and tested on the same domain. This is especially true for legal.

The second part of table 3 shows the models that have been jointly trained on two domains. The third part of table 3 shows the results for the models that are fine-tuned on legal and have been adapted to one of the target domains. It shows that the models trained on two domains jointly in general outperform the models trained with active learning when looking at the source domain. This indicates that when we apply active learning we are not able to maintain the performance on the source domain.

C Group Contributions

The contributions stated below indicate who had the main responsibility for each part. However, multiple group members contributed to every part of the project.

- Related work: Ida
- Data cleaning (converting to BIO-tagging): Sebastian
- Active learning: Ida and Joke
- Model training on the HPC: Andreas
- Model evaluation: Andreas
- Data Map: Joke
- Preprocessing data (NERutils): Andreas
- Visualizing results: Sebastian and Andreas

D Usage of chatbots

We have sporadically consulted generative AI technologies for technical assistance, e.g., when writing code to create plots for the report.

Models	**Finetune [✓] / Active Learning [x]**							**Method**				**Macro span-F1**							
	News	SoMe	Conv.	Legal	Web	W&B	dannet	Random	Margin	Confid.	Vocab	News	SoMe	Conv.	Legal	Web	W&B	dannet	All
News	✓											**0.93**	0.59	0.64	0.50	0.66	0.65	0.67	0.64
SoMe		✓										0.70	**0.79**	0.63	0.50	0.59	0.51	0.73	0.59
Conversation			✓									0.74	0.65	**0.82**	0.55	0.60	0.57	0.73	0.60
Legal				✓								0.87	0.75	0.65	**0.92**	0.67	0.64	0.57	0.71
N+SoMe	✓	✓										**0.90**	0.79	0.65	0.54	0.67	0.61	0.73	0.66
N+Conv.	✓		✓									**0.89**	0.62	0.83	0.61	0.62	0.58	0.73	0.63
N+Legal	✓			✓								0.92	0.73	0.71	**0.93**	0.66	0.68	0.47	0.71
N+SoMe+R	✓	x						✓				0.74	0.79	0.59	0.46	0.54	0.50	**0.80**	0.54
N+SoMe+M	✓	x							✓			**0.89**	0.80	0.64	0.51	0.63	0.58	0.57	0.62
N+SoMe+C	✓	x								✓		**0.87**	0.80	0.64	0.58	0.62	0.55	0.50	0.62
N+SoMe+V	✓	x									✓	**0.87**	0.81	0.67	0.50	0.63	0.59	0.73	0.62
N+Conv.+R	✓		x					✓				**0.89**	0.63	0.81	0.65	0.67	0.66	0.36	0.68
N+Conv.+M	✓		x						✓			0.71	0.56	**0.80**	0.48	0.53	0.54	0.40	0.55
N+Conv.+C	✓		x							✓		0.79	0.58	**0.82**	0.58	0.60	0.61	0.36	0.61
N+Conv.+V	✓		x								✓	**0.84**	0.59	0.82	0.55	0.64	0.62	0.40	0.64
N+Legal.+R	✓			x				✓				0.85	0.68	0.62	**0.89**	0.63	0.62	0.40	0.67
N+Legal.+M	✓			x					✓			0.69	0.72	0.57	**0.88**	0.51	0.50	0.40	0.59
N+Legal.+C	✓			x						✓		0.81	0.73	0.65	**0.88**	0.62	0.61	0.42	0.67
N+Legal.+V	✓			x							✓	0.82	0.67	0.60	**0.88**	0.66	0.62	0.42	0.64

Table 3: Results from different models