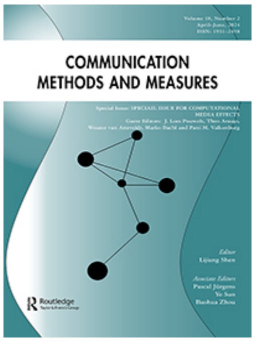


Taylor & Francis
Taylor & Francis Group

Communication Methods and Measures

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/hcms20

Bootstrapping public entities. Domain-specific NER for public speakers

Sami Nenno

To cite this article: Sami Nenno (13 Aug 2024): Bootstrapping public entities. Domain-specific NER for public speakers, Communication Methods and Measures, DOI: 10.1080/19312458.2024.2388098

To link to this article: https://doi.org/10.1080/19312458.2024.2388098

Published online: 13 Aug 2024.

COMMUNICATION METHODS AND MEASURES
https://doi.org/10.1080/19312458.2024.2388098

Routledge
Taylor & Francis Group

Bootstrapping public entities. Domain-specific NER for public speakers

Sami Nenno[a,b]

[a]Centre for Media, Communication and Information Research, University of Bremen, Bremen, Germany; [b]Alexander von Humboldt Institut für Internet und Gesellschaft, Berlin, Germany

ABSTRACT
Named Entity Recognition (NER) is a supervised machine learning task that finds various applications in automated content analysis, as the identification of entities is vital for understanding public discourse. However, sometimes the standard NER labels are not specific enough for a given domain. We introduce Public Entity Recognition (PER). PER is a domain-specific version of NER, that is trained for five entity types that are common to public discourse: politicians, parties, authorities, media, and journalists. PER can be used for pre-processing documents, in a pipeline with other classifiers or directly for analyzing information in texts. The taxonomy for PER is taken from the database of (German) public speakers and aims at low-threshold integration into computational social science research. We experiment with different training settings, involving weakly supervised training and training on manually annotated data. We evaluate multilingual transformer models of different sizes against rule-based entity matching and find that the models do not only outperform the baseline but also reach competitive absolute scores of around .8 and higher in F_1. We further test for generalization and domain adaptation. We show that with only around 100–150 additional sentences, the model can be adapted to new languages.

Introduction

Deploying computational tools for content analysis has become increasingly popular in communication research and political science in recent years. The use of computational tools allows researchers to move from laboratory environments to actual online behavior and "from small-N to large-N" (van Atteveldt & Peng, 2018). One recurring aspect of content analysis is identifying public or political entities, like politicians, parties, or authorities. Recognizing these entities is essential for understanding political dynamics and the public discourse. In this paper, we introduce Public Entity Recognition (PER), a domain-specific version of Named Entity Recognition (NER) that aims at recognizing public entities at the token level. We conduct extensive testing and validation in order to improve the reliability of our models, to provide transparency on possible gender biases, and to document its hardware requirements and its ability for adaptation. While PER is trained on German language data, we show that it requires only few resources for adaptation to new languages and domains. We aim to tackle common issues of computational tools for social science research, such as the prioritization of the English language (Baden et al., 2022) and the scalability and accessibility of these tools (Trilling & Jonkman, 2018). Our models

CONTACT Sami Nenno sami.nenno@hiig.de Centre for Media, Communication and Information Research, University of Bremen, Linzer Str. 4, Bremen 28359, Germany

are available on Huggingface, a well-established platform, in order to allow easy integration into common workflows.[1]

By identifying public entities, PER aims at a very basic task that finds a variety of possible applications in communication research and political science. It can be used for pre-processing text data, for example, by classifying topics for which the entities are used as proxies. Another option is to integrate PER into a larger pipeline of computational methods for content analysis in order to not only identify public entities but to extract more complex information on them (Pilny et al., 2019). However, it can also be used in isolation and for direct application, for instance, to measure individualization, understood as the visibility of political leaders, which can be measured as the ratio of mentions of politicians and parties (Van Aelst et al., 2012).

This work is partly motivated by the introduction of the DBoeS database of public speakers (Schmidt et al., 2023), which lists actors that contribute to the German social public.[2] As (Baden et al., 2022) note, the development of computational methods is often disconnected from social science research. This is, for example, the case if established knowledge from the social sciences is not reflected in the computational tools. We aim at integrating insights from communication research into PER from the start. The DBoeS database is used to guide the data selection, for deriving the entity types of PER, and for weakly supervised training. By incorporating the database into our model, we hope to meet the needs of social scientists.

In this study, we tackle several methodological challenges that accompany NER and weak supervision. Our aim is to build a model that a) works better than simple key-word matching, b) reaches competitive results compared to other domain-specific NER models, c) generalizes rather than memorizes, and d) adapts with only small effort to new domains and languages.[3]

Our strategy is to use continuous fine-tuning (CFT), i.e., fine-tuning in several steps on different datasets and to create step by step a large PER-dataset in a weakly supervised manner and with a bootstrapping approach, i.e., by iteratively using a model to get better and better labels. We evaluate the model on manually annotated data from different domains and languages and explicitly test if the model generalizes successfully and how much new data is required to adapt it to new languages and domains. We find that it is possible to outperform a key-word search by far and to reach an F_1-score of around 0.8 on data from the same but also other domains than the training data. Moreover, with only 100–150 manually annotated sentences, the model can be adapted to new languages, such as English or Spanish, and reach the same scores as on German texts. We end with recommendations for future research to build on our study and to extend our own taxonomy.

Related work

Automated content analysis and named entity recognition (NER)

Content analysis is a key method in communication research and other social sciences and it is increasingly performed with the aid of computational methods. While these methods are often rule-based keyword matching, machine learning (van Atteveldt et al., 2021) and especially transfer learning play a significant role, too (Kroon et al., 2023). One sub-process that frequently occurs in content analysis is the identification of named entities, such as persons or organizations (Pilny et al., 2019). This is little surprising as the dynamics of public discourse cannot be understood without reference to individual actors.

(Balluff et al., 2024) provide a review of multilingual NER tools for the social sciences. They found that despite recent progress in NLP, that is often due to the transformer architecture, most of the reviewed approaches use dictionaries and only few make use of supervised machine learning of which

[1] https://huggingface.co/Sami92

[2] "gesellschaftliche Öffentlichkeit."

[3] The code can be accessed at https://osf.io/4fhze/

transformer models make up only a subset. However, in their experiments they find that XLM-R (Conneau et al., 2020) reaches the best performance and the second-best inference speed. Moreover, due to its multilingual pre-training, XLM-R is able to identify entities in languages for which it was not NER-fine-tuned (Tolochko et al., 2024) conduct experiments to find out about the effect of keeping or removing named entities from corpora on topic modeling. They find that named entities not only influence structural properties of the topic models like the proposed number of topics but also affect the (human) interpretability of the topics. They conclude that keeping named entities can bias the model output with regard to, for example, the association of a politician or party with a certain topic.

However, while the use of automated methods for NER is frequent, there is little research in political science and communication research on developing custom methods for this task.[4] Our discussion of existing methods is therefore mostly limited to research in computer science and natural language processing (NLP).

(Jehangir et al., 2023) describe Named Entity Recognition (NER) as "the process of identifying numerous segments of information referenced in a text and then classifying them into pre-established categories." For classical NER, these pre-established categories are *persons*, *locations*, *organizations*, and *miscellaneous*. NER is a sequence tagging or token classification task. This means that while the input to a NER model is a text span, for instance a sentence or an entire paragraph, the span is not classified as a whole but instead each segment (i.e., a word or sub-word) is assigned its own class label.

The most popular dataset for training NER models is CoNLL-2003 (Tjong Kim Sang & De Meulder, 2003. The data involves several languages, such as English (taken from *Reuter's* news stories) and German (taken from articles of the *Frankfurter Rundschau*). Benchmarking NER models often happens on CoNLL-2003 due to its accessibility. One key limitation, however, is its age. The articles for CoNLL-2003 range from 1996 to 1997. A more recent German language dataset for NER is provided by (Benikova et al., 2014), which follows the normal NER-scheme with four classes but also accounts for nested entities. Another widely used dataset for NER is OntoNotes (Weischedel et al., 2013). The taxonomy of OntoNotes is more fine-grained than that of CoNLL-2003. It involves the same four class types but also 14 additional ones like Date, Event, Law, or Product. An example for a dataset drawn from Tweets is WNUT-17 (Derczynski et al., 2017), which comprises some additional classes and for which current ranks document an F_1 of 0.61 (Wang et al., 2021).[5]

(Yu et al., 2020) provide a comprehensive overview of the scores (F_1) of recent approaches to NER on the different versions of the CoNLL-2003 dataset. On the English version of CONLL-2003, the scores range from 0.916 to 0.935. On the German version, they range from 0.788 to 0.903. For Spanish they range from 0.858 to 0.903 and from 0.817 to 0.937 for Dutch. More recently (Schweter & Akbik, 2021), reach a score of 0.9309 on the German CoNLL-2003 dataset. We take their approach as role-model for our experiments (further specified in Methods-section).

Domain-specific NER

Domain-specific NER can be understood in different ways: Either it is meant as NER for specific document types such as social media, newspaper articles, or novels or as NER with domain-specific class labels such as medical entities, fine-grained locations, or – as for our case – public entities. Often, the second type comes with the first, because domain-specific labels tend to occur more frequently in documents of the same domain. For the present study, domain-specific NER is understood as NER with custom class labels, i.e., NER that classifies public entities rather than the classical entities like persons, locations, etc. However, our aim is to build models that work for various types of documents. Accordingly, we will also turn to domain generalization.

The most extensive approach to domain-specific NER is CrossNER (Liu et al., 2021). CrossNER contains five domains: politics, natural science, music, literature, and artificial intelligence (AI). For

[4] To our best knowledge, there is no research in journals like *Computational Communication Research*, *Communication Methods and Measures*, or *Political Analysis* that is dedicated to developing custom NER models.

[5] https://paperswithcode.com/sota/named-entity-recognition-on-wnut-2017

Table 1. Overview of recent approaches to (domain-specific) NER.

Type	Publication	Dataset	Lang.	Classes	Architecture	F_1
Classical NER	Yu et al. (2020)	CoNLL-2003	EN	4	BERT+LSTM	.94
	Yu et al. (2020)	OntoNotes	EN	18	BERT+LSTM	.91
	Schweter & Akbik (2021)	CoNLL-2003	DE	4	XLM-R	.93
	Wang et al. (2021)	WNUT-17	EN	6	BERT+CRF	.61
Domain-specific NER	Liu et al. (2021)	CrossNER	EN	9	BERT	.73
	Ushio et al. (2022)	TweetNER	EN	7	RoBERTa	.67
	Chen et al. (2022)	HarveyNER	EN	4	BERT	.70
	Kumar & Starly (2022)	FabNER	EN	12	BiLSTM+CRF	.88

each of these domains, they label roughly 1000 sentences from Wikipedia, respectively. For the politics-domain (Liu et al., 2021), use the classical NER categories and add: *Politician*, *Political Party*, *Event*, *Election*, and *Country*. Their maximum F_1-score for this domain is 0.728. For the other domains, it ranges from ~0.63 to ~0.74. Another approach to domain-specific NER comes from (Ushio et al., 2022). Their dataset, TweetNER7, consists of about 11.000 Tweets and the labels are *person*, *location*, *corporation*, *creative work*, *group*, *product*, and *event*. They report a maximum F_1-score of 0.665. Other approaches to domain-specific NER include: Fine-grained locations (Chen et al., 2022) or manufacturing domain (Kumar & Starly, 2022). In line with (Balluff et al., 2024), most of the recent approaches to (domain-specific) NER use the transformer architecture (Table 1).

All of the mentioned approaches to domain-specific NER document strong variance in the performance on different class-labels. For example (Ushio et al., 2022), report an F_1-score of about 0.8 for the category *person* but only 0.4 for the category *creative work*. They note that the performance correlates with the number of training examples per class. They further add that the entity diversity plays a role for the performance: Classes with a higher number of unique tokens tend to be more difficult.

Weakly supervised learning (WSL)

NER is a task that belongs to the field of supervised machine learning (SML). For SML a dataset, e.g., a text corpus, is labeled by annotators and subsequently a statistical model is trained on this data. The aim of SML is to build an automated classifier that performs the same task as the annotators did but on new data. For classical SML, the annotators are (trained) human coders. In contrast, weakly supervised learning (WSL) is an approach to machine learning that uses hard-coded rules or other heuristics (*weak annotators*, in the following) to label a dataset and prepare it for supervised learning. Since human annotations are resource-intensive in terms of time and money, WSL enjoys popularity in low-resource settings. While being resource-efficient, the major drawback of WSL is that the resulting weakly labeled dataset comes with a high noise rate, i.e., many incorrect labels. More formally, let $\mathrm{D_w} = \{(x_i, \widehat{y}_i)\}_{\mathrm{i}=1}^{\mathrm{N}}$ and $D_\mathrm{c} = \{(x_i, y_i)\}_{i=1}^{N}$ be two datasets, whereas the subscript "w" indicates that the labels $\hat{y}$ are derived by weak annotation and the subscript "c" indicates that the labels y are derived via human (clean) annotation.[6] The goal of WSL is to train a model on $\mathrm{D_w}$ and to generalize well on $\mathrm{D_c}$

Recent approaches to WSL use different strategies to achieve this goal. Yu et al. (2021) introduce COSINE. This method uses a set of weakly labeled data for the initial training of a model and in a second step, soft pseudo-labels are derived for additional data using contrastive learning. They test the method on different datasets (including token classification) and report an increased performance when compared to previous baselines. (Liu et al., 2021) use a method in which the unlabeled data is automatically re-labeled over multiple rounds and apply this method to NER. They report an F_1-score

[6] Different disciplines refer differently to these kinds of labels, for example as "ground-truth" or "gold-standard" We acknowledge that these names carry connotations that not everyone accepts. In the following, we refer to human annotations as "clean labels" or "manual labels."

of 0.789 on the CoNLL-2003 dataset. (Lison et al., 2020) follow a different strategy and aggregate labels from multiple weak annotators, such as semantic rules (e.g., matching person names from a database). In a second step, they use a hidden Markov model to derive final labels. They achieve an F_1-score of 0.754 on the CoNLL-2003 dataset. In a later publication, they publish a Python-library, called *skweak*, for WSL that supports their approach (Lison et al., 2021). We used *skweak* for weak annotation for the present study. However, we did not make use of the hidden Markov model and relied on a simple majority vote to derive the final weak labels.

(Zhu et al., 2023) show that most approaches to tackle label noise in WSL are significantly overestimated. Their core argument is that most of these approaches require clean validation data to find the right set of hyperparameters. However, when using this clean data for model training instead of for the respective method to reduce label noise, they achieve competitive or even better results. They show, for example, that for OntoNotes, they require only 400 sentences with clean labels in order to outperform the approach by (Yu et al., 2021). In further experiments, they show that Continuous Fine-Tuning (CFT), i.e., first fine-tuning on weak labels and then on clean labels, does not only improve over simple fine-tuning on clean data but also over most sophisticated methods for WSL. For the present study, we therefore do not use the previously discussed methods but rely on CFT.

WSL comes with obstacles that require consideration. One major danger is that the model fails to generalize and memorize instead. By generalization, we mean that the model learns the common patterns in the task and discards irrelevant noise and outliers (Tänzer et al., 2022). Memorization means that the model does not learn the common patterns but rather the individual entity names and class labels that are provided in the dataset. For WSL, this is particularly damaging because we already have a set of weak annotators that can do that. In other words, in order for WSL to be successful, models need to generalize. Otherwise, we can simply use the weak annotators.

(Tänzer et al., 2022) perform an in-depth study of memorization vs. generalization for pre-trained language models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). They identify three phases during the model training with noisy NER datasets. For the first few epochs, the model generalizes well and the performance increases on both, training and validation data. The next few epochs form a settling phase during which neither training nor validation performance changes. During the last epochs, however, the performance on the training data increases, while the performance on the validation data decreases. They show that this is because the model fails to find a pattern among the noisy labels in the training data, and it starts memorizing instead. This can inform the present study: While training with noisy labels is beneficial even on clean labels for the first few epochs of the training process, it can be damaging during later stages of the training. Accordingly, we can identify the "sweet spot" and interrupt the training at the right time. Additionally, the study by (Tänzer et al., 2022) is a reminder why it needs clean validation data: Strong performance on noisy data is not a reliable indicator for a good model.

Data & annotation

Database of public speakers (DBoeS)

The database of public speakers (DBoeS) lists actors that contribute to the German social public and is provided and regularly updated by (Schmidt et al., 2023).[7] The DBoES comprises three overarching categories: *media*, *organizations*, and *persons*, which are each divided into further sub-categories. For more information on how individual entities were sampled, see the DBoeS-documentation. The database lists names of, for example, politicians, their social media profiles, and further information like gender or party affiliation.

For our purposes, the database is divided into five categories: *media*, *authorities*, *parties*, *politicians*, and *journalists* (see Table 2). This division is motivated by two thoughts. On the one side, we aim for as many categories as possible, in order to build a NER-model that operates on a high level of detail. The most fine-grained classes that are provided by (Schmidt et al., 2023) are *newspaper*, *public broadcast*,

[7] https://github.com/Leibniz-HBI/DBoeS-data/tree/main

Table 2. Entity distribution and description of the per taxonomy that is based on the database of public speakers.

Class	Number	Percentage	Description
Media	2169	30.3	Newspapers, Public/Private Broadcast (news and entertainment), News agency.
Authority	379	5.3	Authorities on national and federal level.
Party	359	5.0	Parties on national and federal level including their youth wings.
Politician	2811	39.2	Members of the German parliament, the federal parliaments, the EU parliament and members of the German government.
Journalist	1451	20.2	Journalists from the federal press conference and other data bases.

news program, *entertainment program*, *online only*, *news agency*, *party*, *authority*, *journalist*, and *politician*. However, at the same time, we take into consideration that each category must receive a sufficient number of individual entities and that the categories should not be too similar (e.g., *newspaper* and *news agency*). We therefore keep the sub-categories of *media* as one class.

Weak annotation

For the present study, weak annotation is mostly performed by matching the names that are provided by the DBoeS database with data from German newspapers and Wikipedia articles. We use only a few additional rules: 1) we automatically add the German *Genitiv*-form to the words, 2) We allow compound words in the form of "Party-X," e.g., "SPD-faction" and manually evaluate if the compound denotes a party or politician, 3) we add often used synonyms for German parties (e.g., "Sozialdemokraten" for "SPD" or "Liberale" for "FDP"), 4) We manually add job descriptions for politicians (e.g., "member of the parliament" or "minister"), 5) We allow surnames to be matched without first names if the full name was already mentioned in the same document (e.g., "Scholz" if "Olaf Scholz" was already mentioned). These rules, together with the entity matching, are referred to as "weak annotators" and are not only used for weak annotation but also as baseline for our models.

For the weakly annotated data, we use two data sources: German newspaper articles (total = 267,786) and German Wikipedia articles (total = 4,348). For the newspaper articles, we scrape all newspapers that are mentioned in the DBoeS database and that are available online between the 25th of September and the 25th of October 2023. We do not scrape articles behind a paywall. For the Wikipedia articles, we scrape all available pages for the DBoeS entities. We assume that due to this even rare entity-instances, like lesser-known politicians, occur in the data.

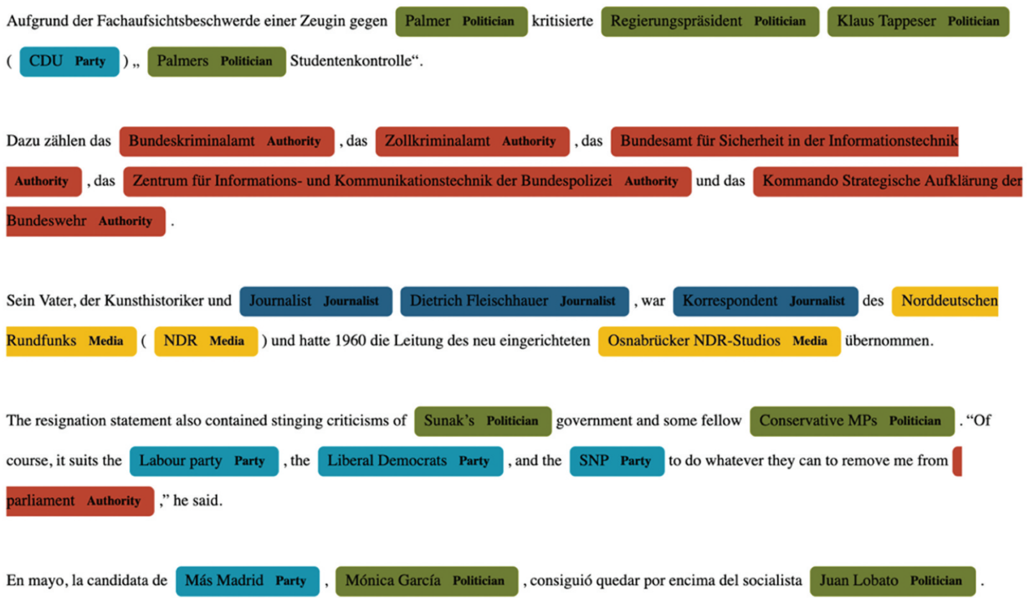
The scraping process resulted in a dataset of about 8.5 million sentences. We sample a subset of about 40,000 sentences for the training process. The sampling is led by the aim to achieve a more balanced class-distribution. However, because we include surrounding words for each sentence (see Methods-section) and because of the irregular occurrence of entities in single sentences, we do not achieve a perfect class-distribution for the subset. Nevertheless, the least frequent class, *journalist*, is now closer to the other classes. Figure 2 displays details on the class-distribution. It also shows the token-count for each entity-class. Tokens for *journalists* and *politicians* are mostly first- and surname or only surname. German words for *authorities* can be relatively long (see Figure 1), while tokens for *parties* are usually only one word.

Table 3 displays the token diversity, i.e., the number of unique tokens per entity class. As expected, *politician* has by far the highest diversity, as it consists mostly of individual names. It is followed by *journalist*, *media*, and *authority*, while *party* has the lowest diversity. Again, this is not surprising because there are only a small number of parties and their names tend to be repeated frequently.

Manual annotation

In order to evaluate the models in different scenarios, we manually annotated data from different sources. The annotation was performed by the author and one student assistant.[8] The coding-

[8] Paid according to collective agreement of Berlin for student employees.



Aufgrund der Fachaufsichtsbeschwerde einer Zeugin gegen Palmer Politician kritisierte Regierungspräsident Politician Klaus Tappeser Politician (CDU Party) „ Palmers Politician Studentenkontrolle“.

Dazu zählen das Bundeskriminalamt Authority , das Zollkriminalamt Authority , das Bundesamt für Sicherheit in der Informationstechnik Authority , das Zentrum für Informations- und Kommunikationstechnik der Bundespolizei Authority und das Kommando Strategische Aufklärung der Bundeswehr Authority .

Sein Vater, der Kunsthistoriker und Journalist Journalist Dietrich Fleischhauer Journalist , war Korrespondent Journalist des Norddeutschen Rundfunks Media (NDR Media) und hatte 1960 die Leitung des neu eingerichteten Osnabrücker NDR-Studios Media übernommen.

The resignation statement also contained stinging criticisms of Sunak’s Politician government and some fellow Conservative MPs Politician . “Of course, it suits the Labour party Party , the Liberal Democrats Party , and the SNP Party to do whatever they can to remove me from parliament Authority ,” he said.

En mayo, la candidata de Más Madrid Party , Mónica García Politician , consiguió quedar por encima del socialista Juan Lobato Politician .

Figure 1. Example sentences in different languages for public entity recognition.

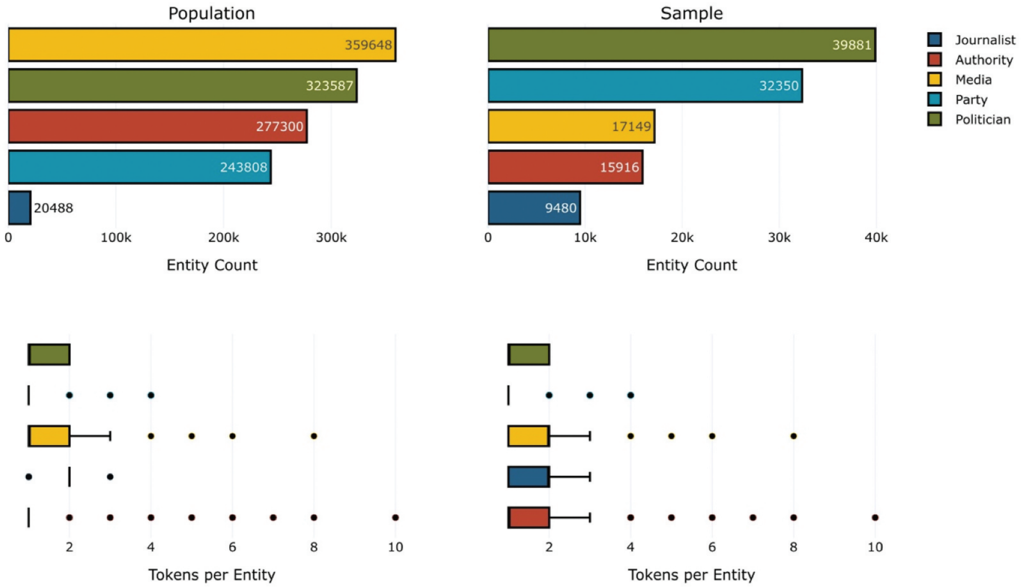


Figure 2. Entity counts and tokens per entity for the weakly annotated data. Left: entire population scraped from newspapers and wikipedia articles. Right: smaller sample that is used for our experiments.

instructions are derived from the DBoeS-documentation by (Schmidt et al., 2023). The annotators reached an F_1-score of .85 on a set of 250 sentences.[9] Afterward, each annotator received their own set of sentences for annotations. Disagreements between annotators followed some patterns. On the one side, there were “common” mistakes like overlooking an entity. However, a recurring disagreement

[9] See the section on evaluation metrics.

Table 3. Entity distribution and other statistics for the weakly annotated sample used for our experiments.

Entity Class	Total number of Tokens	Unique Tokens	Diversity (%)
Politician	39,881	8,501	21.3
Journalist	9,480	654	6.9
Media	17,149	1,054	6.15
Authority	15,916	494	3.1
Party	32,350	932	2.88

Table 4. Size and description of manually annotated datasets.

Dataset	Sentences	Description
In-domain	3090	Roughly same time period and similar topics as weak sample.
Out-of-domain	558	Agenda 2010 (beginning 21st century) and Spiegel-affair (1960s).
Twitter	567	German national election 2021.
English	224	British Prime ministers and attack on US-Capitol.
Spanish	344	Spanish national election 2023 and Catalan independence referendum.

was on whether the country specification of a party mention should be included or not (e.g., "CDU Deutschlands"/"German CDU"). It was decided to include it. Another disagreement was the splitting of compound descriptions, e.g., "Sachsen-Anhalts Wirtschaftsminister und CDU-Chef Sven Schulze"/"Saxony-Anhalt's Economics Minister and CDU leader Sven Schulze." It was decided to code this as one rather than as two entities.

The data was chosen according to different criteria (see Table 4). For the in-domain dataset, we chose data from roughly the same period as for the weakly labeled data. For out-of-domain data, we chose data from the first years of this century and from news articles on the "Spiegel-affair"[10] during the 1960s. For both datasets, we took documents from major German newspapers (e.g., *Spiegel*, *taz*, or *Süddeutsche Zeitung*), governmental websites (e.g., the self-descriptions of German ministries), and documents for political education.[11] We expect that, for example, all or at least most politicians mentioned in the out-of-domain data do not occur in the DBoeS-data.

In order to test performance on a different media format, we also sampled Tweets from the last national election in 2021. We chose the data of the election, as well as the day before and after and sampled tweets with the most frequent topic-related hashtags.[12] Finally, we test how they perform for different languages, as well. We chose English and Spanish. For the English dataset, we sampled news articles from the Guardian. One article for each of the last three prime ministers[13] and additionally, we annotated the English Wikipedia article on the attack on the US-Capitol in January 2021. For the Spanish dataset, we sampled newspaper articles from *El País* on the recent national election in Spain. We also added the Spanish Wikipedia article on the Catalan independence referendum in 2017.

As we deal with naturally occurring texts, we had little influence on the entity distribution, except for the choice of thematic. While the in-domain dataset is relatively balanced, for the other datasets at least one class is either much more frequent or very rare topics (see Figure A1 in Appendix). Assuming that *party* is rather easy to classify, we expect that the results on the Twitter data give us only a distorted picture. In other words, we expect the models to perform strongly on Twitter because a good performance on *party* is already sufficient for it. However, the imbalanced class-distribution is not necessarily a distortion of the real distribution. For example, *journalist* occurs only rarely in almost

[10] https://en.wikipedia.org/wiki/Spiegel_affair

[11] E.g., https://www.deutschland.de/de/topic/kultur/kommunikation-medien/die-zeitungen-im-medienland-deutschland

[12] "btw21," "AfD," "Laschet," "CDU," "Habeck," "Jamaika," "Grüne," "SPD," "Bundestagswahl," "Bundestagswahl21," "wahl2021," "btw2021," "Scholz," "FDP," "btw21," "btw2021," "Baerbock."

[13] Boris Johnson, Liz Truss, and Rishi Sunak.

all datasets and this makes sense, given that journalists are usually the ones who write articles and are not mentioned by articles.

Method

Baselines and bootstrapping weak labels

We apply three strategies for predicting labels. The first are the weak annotators, i.e., rule-based matching of names from the DBoeS data as described in the previous section. We use this as a baseline and also refer to it in the following as such. One of the main questions of this study is if we can train a model that is better in identifying public entities than simple matching. This question will be answered by comparing our models to the baseline. The second strategy is simple fine-tuning, i.e., using a pre-trained model (see next section for details) and fine-tune it on the in-domain dataset. We refer to this approach as FT_c, whereas the "c" indicates that no weak but only clean labels are used for training. Comparing our approach to FT_c indicates if training on weakly annotated data improves the performance. Finally, we apply the approach that was suggested by (Zhu et al., 2023): Continuous Fine-tuning (CFT). In a first step, a model is trained on the weakly labeled data. We refer to this as FT_w. In a second step, the model is fine-tuned a second time but this time on the clean in-domain dataset.

We further use a bootstrapping approach to increase the label quality of the weakly labeled data. Because CFT is able to detect new entities that were not identified by the weak annotators, we use the model to re-annotate the weakly labeled dataset. The process consists of the following steps: 1) split the weakly annotated data into training and test set using 5-fold cross-validation (which results in an 80:20 split), 2) train the model on the training set, 3) further fine-tune on in-domain data, 4) use the CFT model for inference on the weakly annotated data, 5) repeat this for all five folds, 6) locate the mismatches between the CFT predictions and the weak annotations, 7) manually[14] inspect the mismatches and find the correct entities that were not found by the weak annotators but by CFT, 8) replace the old labels by the new labels. This process is repeated several times. We refer to each as an iteration. Iteration 0 denotes the original weakly labeled dataset for which we only use the weak-annotators. Iteration 4 is special because we do not apply the bootstrapping method to the previously described sample of about 40,000 sentences but to a new sample of unlabeled data. This data comes from the same newspaper corpus from before but between the 25th of October 2023 and 25th of November 2023. This sample is much larger and consists of about 200,000 sentences each containing at least five entities. For iteration 4 we first fine-tune on this large sample and then again on the smaller sample of 40,000 sentences.

Model architecture and hyperparameter settings

For our experiments, we use XLM-R (Conneau et al., 2020) in its Base and Large version. XLM-R is a multilingual pre-trained transformer for 100 languages. It is therefore applicable to German text but also to text from other languages. We choose this architecture on the one side because of its applicability across languages and on the other side because it is well tested for applications in computational social sciences (Kroon et al., 2023; Laurer et al., 2024) and in the domain of NER. (Schweter & Akbik, 2021) use this model for their NER experiments for German, and we take over some of their hyperparameter settings. We choose a learning rate of 5e-6 and the "Reduce learning rate on plateau"-scheduler as it is implemented in *Pytorch*. This scheduler reduces the learning rate when a chosen metric, F_1 in our case, stops improving.[15] (Schweter & Akbik, 2021) use a batch size of 4.

[14] Manual inspection was conducted by the author.

[15] Note that (Schweter & Akbik, 2021) use "One-cycle LR" as scheduler in their publication but "Reduce learning rate on plateau" in the current implementation of the Flair-library: https://github.com/flairNLP/flair/blob/master/flair/trainers/language_model_trainer.py (accessed 7th Dec. 2023).

However, we notice that a batch size of 8 speeds up training and has no impact on performance. Furthermore, while (Schweter & Akbik, 2021) train for 20 epochs, we only train for 5 epochs on the weak labels because our dataset is much larger and the performance stopped improving for more epochs. For FT_c and CFT, we train for 15 epochs. Finally, we take over their approach to include a context-window of 64 words to the left and right of a sentence. In other words, each individual training/test example consists of a sentence and its 128 surrounding words (we keep referring to it as a sentence).

Labeling scheme & evaluation metrics

We use BIO-tagging for the annotation as it is standard in NER. B-XXX is assigned to the first token of an entity, I-XXX is the label for subsequent tokens of the same entity, and O is assigned to all tokens that do not correspond to an entity. We assume that entities are non-overlapping. We evaluate the mismatch between the prediction and the correct label on the BIO-level. In other words, for a correct prediction, the model must not only predict the correct entity class but also the correct BIO-label.

Performance is measured in (weighted) precision, recall and F_1 and in accordance to the standards set by the CoNLL task. We use the *sequeval*-library for computing all scores.[16] Precision is understood as the ratio of true positives (TP) and the sum of TP and false positives (FP). Recall is understood as the ratio of TP and the sum of false negatives (FN) and TP. F_1 is understood as the harmonic mean of precision and recall. Recall indicates if the model finds all relevant entities, precision indicates that the model is more careful in its predictions, and F_1 indicates a balanced relation between precision and recall. All reported results are averaged over 5-fold cross-validation with the exception of the experiments on language adaptation, for which we used train-test-splits with different random seeds.

Testing memorization

As mentioned before, one danger of weak supervision is that the model memorizes rather than generalizes. Following (Tänzer et al., 2022), we speak of generalization if the model learns the common patterns in the task and discards irrelevant noise and outliers. In contrast, memorization means that the model does not learn the common patterns but rather the individual entity names and class labels that are provided in the dataset. We assume that the weak labels are relatively noisy compared to manual labels. Label noise is likely to obscure the class-specific patterns that the model is supposed to learn and accordingly, the chance of memorizing is increased.

Memorization can be understood as a prediction for a token for which the most important token is the given token itself. In order to measure the importance of each token in a sentence, we use Integrated Gradients (IG) (Sundararajan et al., 2017). IG is a method for attributing the prediction of a deep network to its input features. In our case, this means attributing the prediction of XLM-R to the individual tokens of a given sentence. More formally, given an input vector $x = (x_1, \ldots ,x_n)$, IG assigns an attribution score $a = (a_1, \ldots a_n)$ to the vector x, whereas each element of a indicates how strong the corresponding element in x contributed to the prediction. We define a memorized prediction as a prediction for x_i in which a_j is the highest attribution score and i=j. In other words, the token itself was most important for the prediction of its label.

In order to test memorization, we create an artificial version of a subset of the weakly labeled dataset. More concretely, we sample a subset of 15,000 examples of the weakly labeled dataset. In a second step, we assign random labels to the entities. For example, the two tokens "Olaf" and "Scholz," which is originally labeled as *B-Politician* and *I-Politician* become *B-Media*, *B-Authority* or *B-Party*, and *I-Party*. The only constant for this random labeling scheme is that identical tokens receive identical labels, e.g., if "Scholz" is labeled as *B-Media* all instances of "Scholz" are labeled as *B-Media*. With this approach we create a dataset on which a model can only perform well if it memorizes. In

[16] https://huggingface.co/spaces/evaluate-metric/seqeval

total, we create five versions of this dataset for which we shuffle 0%, 70%, 80%, 90%, or 100% of the labels randomly.

Our hypothesis for this experiment is that memorization is stronger for the random datasets than for the dataset with 0% random shuffling. This would indicate that the model memorized less on the original dataset and rather generalized. We measure three variables: 1) The F_1-score of a model that is trained on the different versions of the dataset, 2) the number of entities for whose classification the token itself had the strongest attribution score, and 3) the average strength of this attribution score. We assume that a) the F_1-score decreases with increasing shuffling, b) the number of predictions for which the token itself is most important increases with increasing shuffling, and c) that the average attribution score for these memorized predictions increases with increasing shuffling.

Results

Weakly supervised training

Figure 3 displays the training curve for FT_w[17] for the first iteration on the weakly labeled dataset and the performance on the manually labeled test datasets. The performance on the weakly labeled hold-out is much higher than that on the manually labeled datasets. As assumed, the performance on Twitter data is strongest and followed by in-domain data. However, it is noteworthy that all curves follow roughly the same pattern. As (Tänzer et al., 2022) observed, models tend to overfit on noisy data, which leads to decreased performance on the manually annotated datasets. This did not happen in this case. Increased performance on the weakly labeled data did not happen at the expense performance on the clean data. Nevertheless, a trade-off between recall and precision is clearly visible. Throughout the training, recall on all datasets is higher than precision and has less variance across datasets than precision.

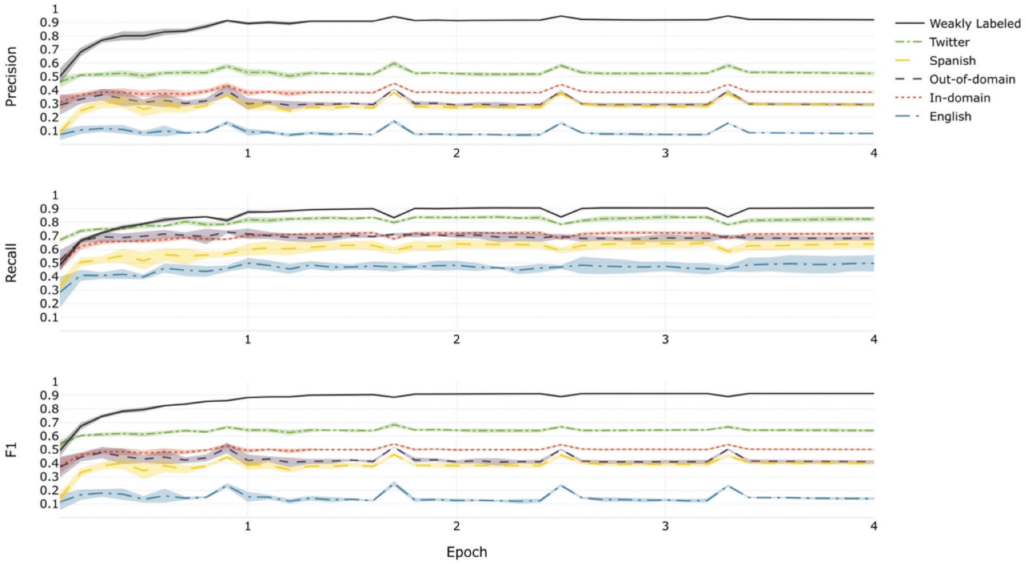


Figure 3. Training curve for XLM-R base on weakly annotated data (FT_w) for iteration 0. Evaluation on weak test data and clean datasets was performed ten times each epoch.

[17] In our discussion, we focus on the base model even though the large model performed stronger throughout. However, because the base model requires less computational resources, we assume that it can be applied easier.

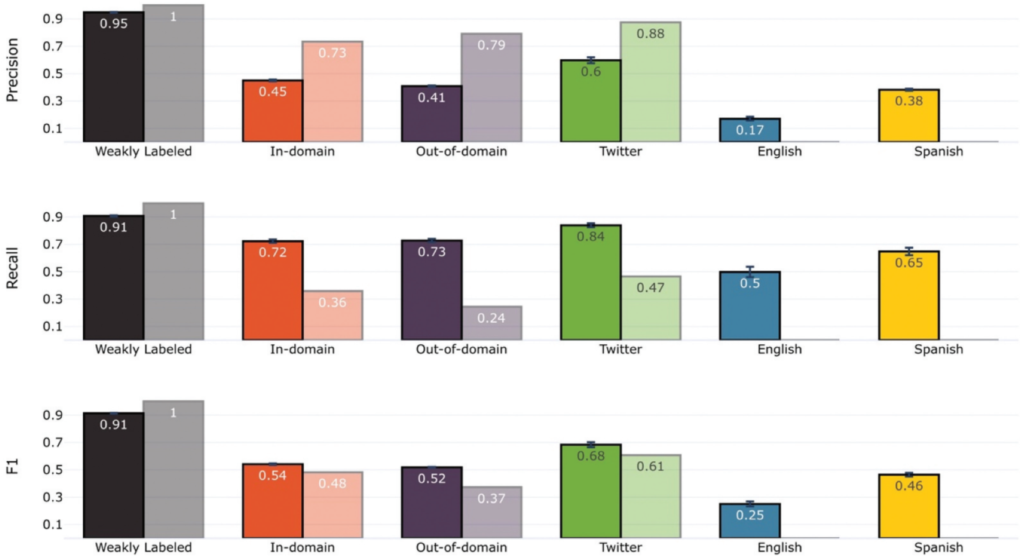


Figure 4. Best scores of FT_w base on iteration 0. Transparent bars represent the baseline of weak annotators.

Figure 4 displays the maximum scores on each of the test datasets for FT_w for the first iteration. The transparent bars display the scores of the weak annotation baseline. As the baseline and the weak annotations are one and the same, the score is 1 for the weakly labeled test dataset. For all other datasets, FT_w beats the weak baseline in terms of F_1. For the English and Spanish data, this is the most obvious as the weak baseline fails to detect any entity. For the data from Twitter, the model performs strongest but also the baseline does better than on the other manually labeled datasets. This is most likely due to the high frequency of Party, which is a category that is relatively easy to learn for FT_w but also relatively easy to match with a rule-based approach.

It is noteworthy that the baseline is superior on all German datasets with regard to precision but not with regard to recall. In other words, XLM-R learns new entities even though it is trained only on the weakly labeled entities and it tends to find more entities than there are, i.e., it has a high false positive rate. This motivates our bootstrapping approach. As the model learns more entities than the baseline identifies, we can use it to find more entities and re-label our dataset.

The detailed scores for FT_w for all iterations can be found in Table 5. With the exception of iteration 2, all iterations have a positive effect on the performance. Since iteration 2 does not lead to improved

Table 5. Overview on (weighted) F_1, recall, and precision of FT_w for all iterations. ID = In-domain, OD = out-of-domain, tw = twitter, en = English, and sp = Spanish.

		F_1					Recall					Precision				
Model	Iteration	ID	OD	Tw	En	Sp	ID	OD	Tw	En	Sp	ID	OD	Tw	En	Sp
FT_w Base	0	.54	.52	.68	.25	.46	**.72**	.73	**.84**	.50	**.65**	.45	.41	.60	.17	.38
	1	.60	.62	.74	.46	.48	.63	.68	.80	.59	.57	.59	.58	.71	.40	.47
	2	.41	.47	.66	.35	.21	.40	.51	.72	.47	.23	.45	.47	.65	.33	.24
	3	.62	.66	.74	.50	.50	.63	.69	.79	.55	.56	.63	.64	.72	.47	.52
	4	**.69**	**.71**	.78	.55	**.55**	.68	**.76**	**.84**	.66	.55	.69	.66	.75	.49	**.54**
FT_w Large	0	.52	.51	.69	.23	.43	**.72**	**.76**	.82	.54	.63	.43	.41	.60	.15	.32
	1	.56	.61	.76	.49	.47	.62	.69	.80	.59	.57	.53	.60	.73	.42	.41
	2	.40	.49	.70	.41	.25	.40	.54	.72	.45	.24	.43	.47	.68	.43	.27
	3	.63	.64	.76	**.56**	.52	.63	.72	.80	.60	.56	.63	.66	.74	**.55**	.49
	4	.68	.63	**.79**	**.56**	.53	.68	.72	.82	**.69**	.59	.69	.56	.78	.50	.49
Baseline	-	.48	.37	.61	.0	.0	.36	.24	.47	.0	.0	**.73**	**.79**	**.88**	.0	.0

results, we adapt our strategy for iteration 3. Instead of using only XLM-R Base for re-labeling, we add predictions of XLM-R Large and take the average over the predicted probabilities. For iteration 4, we use a larger dataset (as described in the previous section). We can make two observations. Firstly, with the exception of iteration 3, all iterations lead to better F_1 and recall scores than the baseline. The improvement in F_1 over the course of all iterations is mostly due to improved precision. This means that the models improve mainly by being more careful in their predictions, while still identifying more entities than the baseline.

Continuous fine-tuning

We use Continuous fine-tuning (CFT) to further improve the model performance. Note that CFT is only performed on in-domain data. No other clean dataset is used for training but only for evaluation. Nevertheless, CFT leads to a strong increase in performance not only on in-domain data but on all other datasets, too (see Figure A2 in the Appendix). The exact results for CFT can be found in Table 6. The highest scores are achieved by CFT with XLM-Large. It reaches an F_1-score of .82 on the in-domain data and Twitter, .79 on out-of-domain data and .62 and .61 on the English and Spanish data, respectively. While the best scores regarding recall are also achieved by XLM-R Large, the best scores regarding precision are more scattered across the different experiments. This indicates that in some cases higher recall is only achieved at the expense of precision. While XLM-R Large performs stronger throughout all experiments, XLM-R Base is often only a few points behind.

We also test fine-tuning only on clean data without prior fine-tuning on weak data and refer to it as FT_c. Prior fine-tuning leads to an improved performance in all scenarios (Figure A2 in the Appendix). CFT achieves a better F_1-score on all datasets compared to FT_c. In fact, in some scenarios even FT_w achieved stronger results than FT_c. This effect is especially visible for datasets of domains that are more distant to in-domain, like Twitter and the non-German datasets. This suggests that the previous fine-tuning step on weakly labeled data is especially helpful when it comes to domain adaptation. This is also indicated by recall and precision for FT_c. Recall is weaker than precision, this means that for FT_c the difficulty is rather to find the previously unseen entities and not so much that its classification is not sufficiently careful. Finally, we can observe that with exception of iteration 2, the improvement of CFT over FT_w decreases in later iterations. This indicates that the increased accuracy of the weak labels decreases the gain from training on clean labels.

With respect to the individual entity-classes, we can make three observations (Figure A3 in the Appendix): 1) The correct predictions outnumber the false predictions by far. However, this

Table 6. Overview on (weighted) F_1, recall, and precision of continuously fine-tuned XLM-R (CFT) for all iterations. FT_c is XLM-R fine-tuned only on clean data but not on weak data. ID = In-domain, OD = out-of-domain, tw = twitter, en = English, and sp = Spanish. Highlighted values indicate highest score for the respective dataset.

		F_1					Recall					Precision				
Model	Iteration	ID	OD	Tw	En	Sp	ID	OD	Tw	En	Sp	ID	OD	Tw	En	Sp
CFT Base	0	.77	.72	.72	.49	.59	.75	.73	.72	.55	.62	.82	.75	.78	.48	.61
	1	.77	.73	.76	.51	.53	.74	.73	.76	.53	.52	.82	.75	.81	.54	.56
	2	.76	.73	.76	.53	.51	.73	.73	.79	.57	.51	.82	.75	.78	.53	.54
	3	.78	.75	.77	.54	.57	.75	.74	.80	.56	.59	.82	.77	.80	.56	.63
	4	.80	.75	.80	.56	.61	.78	.78	**.84**	.64	**.67**	.84	.74	.79	.53	.61
CFT Large	0	.79	.76	.75	.54	**.63**	.76	.77	.76	.60	.60	.83	.77	.78	.51	**.67**
	1	.78	.77	.78	.57	.60	.75	.77	.78	.59	.60	.83	.78	.81	**.60**	.63
	2	.79	.77	.75	.59	.59	.75	.77	.75	.63	.59	.84	**.81**	.78	.59	.63
	3	.79	.78	.79	.60	.62	.76	.78	.82	.66	.62	.84	.79	.79	.57	.64
	4	**.82**	**.79**	**.82**	**.62**	.61	**.80**	**.80**	**.84**	**.73**	.66	**.85**	.78	.82	.57	.60
FT_c Base	-	.73	.69	.68	.45	.52	.71	.69	.70	.43	.49	.81	.77	.72	.54	.63
FT_c Large	-	.75	.72	.70	.48	.60	.73	.71	.69	.51	.61	.82	.78	.71	.53	.66
Baseline	-	.48	.37	.61	.0	.0	.36	.24	.47	.0	.0	.73	.79	**.88**	.0	.0

was already clear from the scores. 2) The major source of confusion is between entities and non-entities (“O”). This goes in both directions: false positives but also false negatives. This is reflected by the scores, as well. Precision and recall do not differ by much. 3) The main source of confusion between individual entity classes is between *politician* and *party* on the one side and *media* and *journalist* on the other. This is surprising as we assumed that *politician* and *journalist* would be a main source of confusion as both mostly consist of personal names. However, it seems that the model rather confuses frequently co-occurring entity categories. Often documents or paragraphs that mention politicians also mention parties and the same is true for journalists and media.

The strongest classes for all XLM-R versions regarding F_1 and for in-domain and out-of-domain are *politician* and *party* (Table A1 in the Appendix). While this was expected for *party*, it was not so clear for *politician*. *Party* comes with many entities, few unique entities, and a low number of tokens per individual entity. For *politician* the number of entities is also high but so is the number of unique entities. And while the former should have a positive effect on the results, the latter is expected to have a negative impact. The weakest class for in-domain and out-of-domain is *authority*. This is true for all versions of XLM-R but not for the baseline. While XLM-R outperformed the baseline with regard to all categories, *authority* is not the weakest category among the baseline results. This can be an indicator that XLM-R struggled with entities that have a high number of tokens (as it is typical for authorities), while rule-based matching is less affected of it.

For Twitter, the performance of CFT and FT_c shows different patterns. The strongest categories are *politician*, *party*, and *media* but FT_c did much worse on *media* than CFT. For English, it is again *politician* and *party* for which the models performed strongest and, for Spanish, the performance on all classes with the exception of *authority* is relatively balanced. This is partly due to the datasets because Spanish is much more balanced than English or Twitter. However, often the strongest class for Spanish is *journalist*. This is not representative as there are only three occurrences of this class in the entire dataset.

Qualitative error analysis

There are some common error types for individual datasets but also across datasets. One common misclassification is that of noun markers for media and parties, e.g., *die* Grünen (*the* Greens) or *der* Spiegel (*the* Spiegel). While the model correctly classifies the name of the party or newspaper, it often fails to classify the article in front of the name correctly. Another source of errors are personal names that do not denote journalists or politicians but other types of persons. Often the model falsely classifies these names as belonging to one of the two categories. Ambivalent entities are also difficult for the model. For example, “Grüne” can denote the Green-party but also a politician from the Green-party. In some cases, the model does not find the correct class for the particular context. As mentioned before, frequently co-occurring entities are mixed up, too. “Stern” and “Spiegel Online,” which are German newspapers, are sometimes misclassified as Journalists. Another case of confusion due to co-occurrence is the classification of a city name as Politician. Often political job descriptions are preceded by a city or country as in “Bremer Bürgermeisterin” (mayor of Bremen) or “deutscher Bundeskanzler” (German chancellor). Some of the tokens that are classified as Politician by the model, are just mentions of a city or country without reference to any politician.

Finally, there are some specific sources of error to the datasets from other domains or languages. Little surprising, one major source of error are names of politicians that are not in the DBoeS data, such as Obama, Trump, or Willy Brandt. A similar source of error is political job descriptions in non-German, e.g., “president” instead of the German “Präsident.” As the training data is exclusively German, the model has problems handling non-German words. Also, some country-specific writing styles lead to errors. For example, the English Wikipedia article on the attack on the capitol often refers to a republican from North-Carolina as R-NC or to a Democrat from California as D-CA. These party abbreviations are often classified incorrectly.

Memorization

The performance with 0% shuffling is by far the highest (Figure A4 in the Appendix). The model quickly increases its performance during the first epoch and then slowly increases the performance over the later epochs. For the shuffled datasets, the increase in performance is slower and evenly spread over the entire training procedure. For the shuffled data, the performance for the earlier epochs is negatively correlated with the percentage of shuffling, i.e., the performance is weaker for data with higher amounts of random labels. For the later epochs, the performance for the different levels of shuffling converges.

Our first assumption is partly confirmed, shuffling leads to decreased performance (Table A2 in the Appendix). However, contrary to our expectations, increased shuffling does not lead to decreased performance but even to slightly increased performance. Our second and third assumptions are confirmed for the experiments for 0%, 70%, and 80% of the shuffled labels: The share of memorized predictions on the total amount of predictions (slightly) increases with increased shuffling. However, if we look at the full picture, our hypothesis is not confirmed because for 90% and 100% shuffling the share of memorized predictions and the attribution scores decreases again.

Language & domain adaptation

The scope of this study is not only the German language and public context but is supposed to generalize across domains. Previous experiments show that even though the models are only trained on German language from a limited period of time, they are able to identify entities in other settings, too. However, for English and Spanish the F_1-scores are a bit over .6 at best. This is not reliable enough for actually applying the model in these languages. Accordingly, we test how much data is necessary to improve the performance to a similar level as for in-domain. The results can be found in Figure 5. For these experiments we fine-tune CFT from iteration 4 repeatedly and each time with more sentences. We can make two observations. 1) With about 100–150 sentences we can achieve a performance close to that on in-domain data. For Spanish, it needs 150 sentences to get an F_1 of .8 and for English only 120 sentences.[18] 2) Precision is slightly higher than recall. This makes sense because almost all entities in this language are new for the model and the problem is rather identifying them all and less classifying too many entities.

We further asked if there are certain features of the data that are relevant for domain adaptation. As previously suggested, a large count of entities could be beneficial, while a large count of unique entities might have a negative impact on the performance. Moreover, entities that consist of many tokens might be more difficult to learn for a model. We test all these hypotheses. The results can be seen in Figure 6. The plots at the top display the results for the English dataset and the ones at the bottom for the Spanish dataset. We measure the number of entities per class, the number of unique entities per class, and the average token count per entity for all subsets of the training data that is used for Figure 5 and compute the Pearson correlation with the F_1-score. However, while in some cases the variables have a moderate influence on the performance of the models, in most cases the influence is only small. For the English data, the correlation is a bit higher than for the Spanish data, but this is likely due to the imbalanced class distribution of the dataset. In total, it seems as if the main lever to improve domain adaptation is to add sentences to the training data, while the count of entities, unique entities and tokens per entity have only a minor impact.

Gender bias

We test the performance of the models on in-domain data with regard to the gender of different entities.[19] Of all politicians and journalists in the DBoeS database, 67% are male and 33% are female. It is therefore possible that the models perform better on male entities. For the German language and in

[18] For XLM-R Large it needs 150 sentences for an F_1 of .82 for Spanish and 140 sentences for .82 for English.

[19] Thanks to an anonymous reviewer for pointing this out.

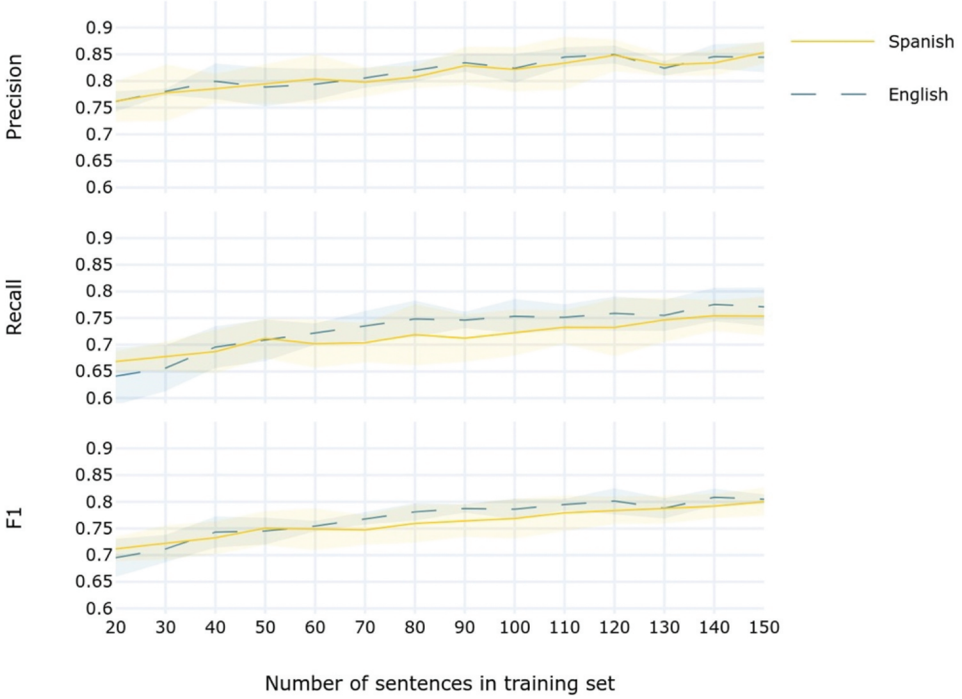


Figure 5. Test results of CFT (base) with an additional fine-tuning step on English/Spanish data. Each point on the x-axis represents a single training run with increasing amounts of sentences.

this context, gender is indicated by either the personal name[20] (Thomas, Sabine) or how the profession is written ("Minister"/"Ministerin," "Journalist"/"Journalistin"). For the in-domain dataset, female entities make up 30% and male entities 70%. Following (Mehrabi et al., 2020), we define three types of error. Due to the low frequency of *journalist*, we do not perform separate experiments for the two person-classes. Let x be a word of class *journalist* or *politician* and j*, p*, o* the predicted label *journalist*, *politician*, or non-tagged.

(1) Type-1 error: $x \neq j^*$ and $x \neq p^*$
(2) Type-2 error: $x \notin \{j^*, p^*\}$
(3) Type-3 error: $x = o^*$

All error types are computed as frequencies in percentage over all x. Type-1 describes misclassifications, i.e., x received an incorrect label or was not tagged at all ("O"). Type-2 describes entities that denote persons but which are not predicted as persons, i.e., as either journalist or politician. Type-3 denotes errors in which x is not tagged at all. The results can be found in Table 7.

For all three types, the error is slightly higher for female entities. For entities for which the gender was identified based on personal names, the error types are a bit higher for male names, and with regard to gender-specific versions of professions the error rate is much higher for the female-versions. The error is about the same for all three types. This is because the main source of errors is entities that qualify as *journalist* or *politician* but are not tagged at all by the model ("O"). We conclude that our models perform weaker on female entities that are referred to via their profession. In general, however, we observe only a small gender bias.

[20] We did not encounter gender-neutral names.

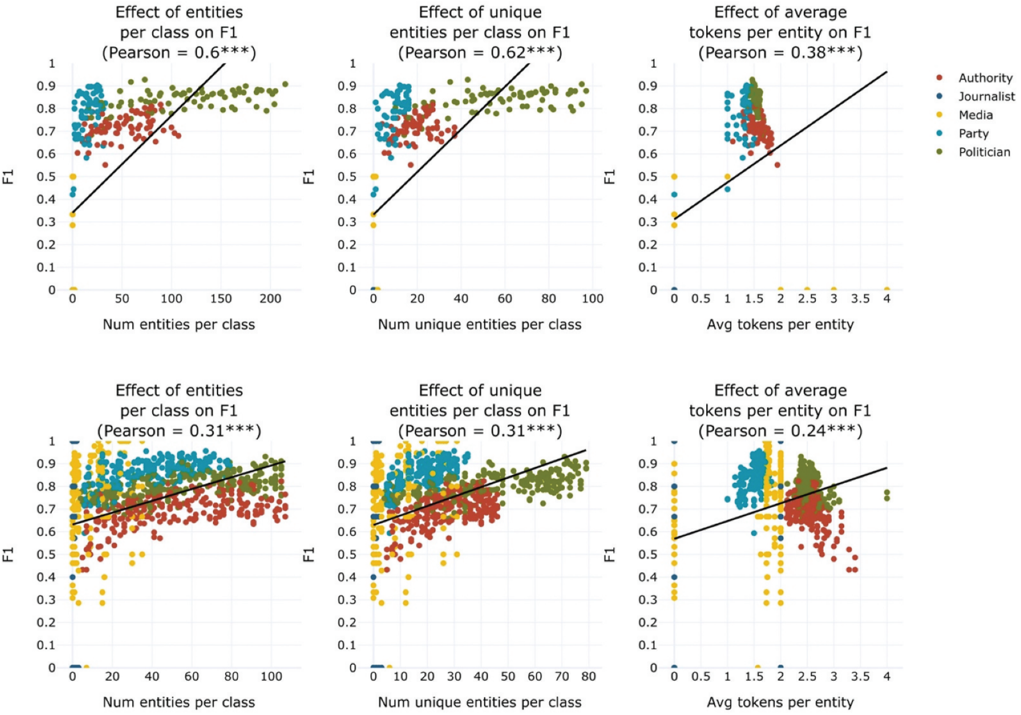


Figure 6. Scatter plot of English (top) and Spanish (bottom) datasets and results on domain adaptation. Each point represents the F_1-score and a) number of entities, b) unique entities, or c) average tokens per entity for one class of one of the training runs with increased number of sentences.

Table 7. Error types with regard to gender for XLM-R base, weakly fine-tuned and then fine-tuned again on in-domain data with 5-fold cross-validation. The first columns display the count of male/female entities (word-level) when they are identified either by name, profession, or both. The remaining columns display the frequency in percentage of the respective error type with regard to gender.

	Count		Type-1		Type-2		Type-3	
	m	f	m	f	m	f	m	f
Name	332	116	**.08**	.07	**.08**	.06	**.08**	.06
Profession	101	69	.1	**.17**	.1	**.17**	.1	**.17**
Both	433	185	.09	**.11**	.09	**.1**	.08	**.1**

Model compression via quantization

In the last round of our experiments, we test how quantization affects the performance of our models. Our aim is to make the models widely accessible and this includes scenarios with limited computing capacities (Trilling & Jonkman, 2018). Quantization is a model compression technique.[21] In the standard setting, model parameters are represented in floating point 32 (FP32) format. For quantization, the precision of the data format is decreased in order to reduce the model's memory footprint. Possible representation formats are floating point 16 (FP16), integer 8 or 4 (INT8/INT4). In theory each compression step should cut in half the memory footprint of the previous step. In practice, this is often not achieved because not all weights and activations can be represented in a lower data format (Dettmers & Zettlemoyer, 2023; Dettmers et al., 2022).

[21] Note that model compression also leads to a decreased electricity usage, which in many cases leads to a reduced carbon footprint.

Table 8. Hardware requirements and F_1-scores for XLM-R with different levels of datatype-precision (tests performed on nvidia tesla V100S-PCIE-32GB). Inference speed refers to processing a single sentence of 256 tokens.

Model	Quant.	Memory (MB)	vRAM (MiB)	Inference (ms)	ID	OD	Tw	En	Sp
CFT Base	FP32	1109.9	7824	**.49**	.79	.73	.80	.56	.59
	FP16	554.9	**4494**	.50	.79	.73	.80	.56	.59
	INT8	470.0	5098	4.26	.79	.73	.79	.56	.59
	INT4	**427.5**	5942	1.21	.74	.68	.75	.51	.53
CFT Large	FP32	2235.4	18488	**.96**	.80	.77	.79	.61	.59
	FP16	1117.7	**10478**	.97	.80	.77	.79	.61	.59
	INT8	815.7	11222	9.24	.0	.0	.12	.0	.0
	INT4	**664.7**	13812	2.72	.79	.75	.80	.57	.60

One major drawback of quantization is that it can affect the model performance. But because this does not necessarily happen, it needs empirical testing. The results of these tests can be found in Table 8. The experiments on the hardware requirements are conducted on a Nvidia Tesla V100S-PCIE -32GB GPU with batch size of 16 and fixed sentence length of 256 tokens. We use the *bitsandbytes*-library as it is integrated in the *transformers*-library.[22] While the reduction to FP16 had no impact on the performance, reducing to INT8 and INT4 had a negative effect.[23] Furthermore, FP16 has the lowest vRAM requirements. This is what is most important for practical application. In most scenarios, memory only plays a role when loading Large Language Models (LLMs) and it does not matter for comparatively small models like RoBERTa or BERT. However, vRAM is often a bottleneck even for smaller models as it forces applications with less computing resources to reduce the batch size to a minimum.

Discussion

Public Entity Recognition (PER) is the task of identifying politicians, parties, authorities, media, and journalists on a token level. It can be used to pre-process large amounts of text data according to their relation to one or more of the categories, but it can also be applied to directly measure concepts, like individualization, that are relevant to political and communication research. In this paper, we showed that our models outperform key-word matching and are on the same level as other approaches to domain-specific NER. In particular, our best models reached an F_1 score of around .8 on each dataset. In comparison, the rule-based baseline reached an F_1-score of .37 to .61 on German datasets and failed completely on non-German texts. Results from other domain-specific NER range from .63–.88. Our model joins these ranks but it did not surpass the results that are typical for classical NER. In this area, scores are often higher than .9. However, it would be unrealistic to expect such results because domain-specific NER is a more challenging task than classical NER.

As it turned out, weak supervision without further fine-tuning on clean data is not sufficient to reach good results. Our models did not even reach an F_1-score of .7 on in-domain data. Yet, the models outperformed the weak annotators already in the first iteration and without clean data. In other words, while not in absolute scores, relative to the baseline, weak supervision did a good job. This is also true when compared to simple fine-tuning, as FT_w outperformed FT_c at least on non-German texts. But for an application in practice, a score of less than .7 is certainly not sufficient and as previous work demonstrates, domain-specific NER can reach better results.

As suggested by (Zhu et al., 2023), continuous fine-tuning (CFT), i.e., first fine-tuning on a large amount of weakly labeled data and then fine-tuning on a small amount of manually annotated data, is a successful strategy. CFT outperformed FT_c on all datasets, which shows that fine-tuning on weak data has a positive effect and was not neutralized by the second fine-tuning

[22] https://huggingface.co/blog/4bit-transformers-bitsandbytes
[23] Especially the performance of XLM-R Large in INT8 is decreased. We ran the experiments a second time to confirm the results, however, we are not able to tell if this is a bug in the *bitsandbytes*-library

step. However, the most important contribution of CFT is not the increased scores on in-domain data. While the increased performance on in-domain data is certainly positive, the most relevant contribution of the fine-tuning step on weakly labeled data is that the models become more robust to domain- and language-shifts. Without being trained on non-German data, CFT Base reaches up to .56 on English texts and .61 on Spanish texts, which is an increase of .11 and .09, respectively, in comparison to FT_c. Since FT_w already outperformed FT_c on non-German data in some situations, we attribute the increased robustness to language-shifts to the first fine-tuning step on weakly labeled data.

Previous work on domain-specific NER suggests that performance across different classes can vary strongly. For PER, this is true, as well. An additional factor that complicates general judgments about PER's performance is that the performance differs for different datasets and even for different sizes of XLM-R. For in-domain data, predictions for *authority* are the weakest with a distance of up to .13 (.11 for XLM-R Large) compared to the strongest class. The in-domain performance for the remaining classes is roughly even, with a slight lead of *politician*. For the other datasets, *politician* and *party* perform more or less reliable and for data from out-of-domain and Spanish, *media* also delivers a good performance (at least for XLM-R Base).

While class-specific performance varies regarding many factors, the main source of confusion is relatively stable across all experiments. The models' misclassifications were usually not between different entity types but between entity and non-entity. Confusions between entity classes happened mostly between frequently co-occurring entities but, nevertheless, they were outnumbered by misclassifications between entities and non-entities.

We conclude that when applying PER in practice, it requires special attention to Authority. However, we expect that the predicted distribution among entity classes is close to the real distribution. Finally, for tasks, like measuring individualization, for which only Politician and Parties are relevant, the models deliver reliable performance.

We performed experiments on memorization using Integrated Gradients (IG) (Sundararajan et al., 2017). However, the experiments results were not conclusive. We expected the number of memorized predictions to increase proportionally with the amount of label noise. This was not the case. While the share of memorized predictions is larger for noisy labels, it does not correlate with the amount of noise. The same is true for the intensity of the attribution scores. However, it is clearly visible that random noise had a strongly negative impact on the F_1-score. This indicates that there is some benefit to the non-shuffled labels. This is further suggested by the performance of FT_w on datasets of other domains and languages. If the models simply memorized the weak labels, they would not be able to make correct predictions on previously unseen entities and languages. However, as the qualitative analysis revealed, some common patterns, like country + political profession (German chancellor), led the model to classify countries as politicians even in the wrong contexts. This can be interpreted as an indication that the model memorized very frequent co-occurrences rather than learning the underlying logic. We conclude, however, that while we could not show directly via IG that our models generalize rather than memorize, overall, the performance of the models suggest that they learned general patterns rather than purely memorizing entities.

For all our experiments, the models performed better on non-German texts than the baseline did. However, in absolute scores, the performance is not sufficient for the models to be applied across languages in practice. We therefore tested how many sentences are required to adapt the models to a new language. The results show that it takes about 100–150 sentences in order to reach similar performance as on German data. This (very) roughly corresponds to five or six mid-sized newspaper articles. We assume that this amount of manual coding is likely feasible even for projects with low resources. Identifying public entities in texts is a less ambivalent task than, for example, sentiment or stance detection. And this means that training coders requires less time. Moreover, as it turned out, number of entities, number of unique entities, and token count per entity have only a minor impact on the model performance. This means that it requires less scrutiny when selecting additional data for adaptation.

The major limitation of our experiments is the class imbalance of the manually labeled datasets. As mentioned already, our results on the Twitter data are likely distorted as the entities consist mostly of *party* and this is the class, which is often the easiest for our models. The opposite is the case for *journalist*, as this class rarely occurs in all but in-domain data. These imbalances are a natural side-phenomenon to PER. There are masses of texts about politicians or parties but only a few mention journalists. Moreover, while a single text often makes references to politicians or parties multiple times, mentions of journalists are less frequently repeated. This means that for sampling sentences with journalists, it is necessary to find many different documents, whereas each contributes only little to the overall count.

Another limitation is the amount of manually labeled sentences. While we annotated more than 3000 sentences of in-domain data, the other datasets consist of only a few hundred sentences. This was due to limited annotation resources and must not necessarily be a limitation. As it was shown, fine-tuning a model can already work with a couple of sentences. However, due to the limited annotation budget, we focused on texts on political topics. This means that we did not test the false positive rate on texts that most likely mention no public entities at all, like the sports-division of a newspaper. We therefore emphasize that our results might not be representative for documents with entity-distributions that are very different from ours.

The last limitation that we want to mention is the difficulties of training and evaluating models across languages. Evaluation on data from different languages usually has two aspects: On the one side, we test how the model performs with regard to different languages and on the other side, we test how the model performs with regard to different domains, as texts from different languages usually come from different nations and therefore with different topics and framing. In our experiments, we did not untangle these aspects and tested for language and domain adaptation simultaneously. One way to untangle these aspects is anchoring, i.e., using parallel corpora (see Licht & Lind, 2023). EU-documents, for example, are often translated to multiple languages while being about the same subject. However, evaluating language and domain adaptation individually for different national contexts requires more annotated data.

Conclusion

In this paper, we introduced a domain-specific NER model for the classification of public entities in documents. Our experiments show that the model does not only performs well on data similar to its training data but adapts across domains and languages. We further showed that it can be fine-tuned for custom use-cases with only small datasets and it can also be deployed in settings with limited computational resources.

Weak supervision had a very positive effect on the model performance, but it came with some practical difficulties. On the one side, it was challenging to make the DBoeS taxonomy work for all possible data sources. The taxonomy covers, for example, politicians with a seat in parliament. But how about political actors who are running for office? While it is probably less important whether to include them or not (we did) and more important to make a choice and stay coherent, this (and related problems) is an issue that might come up only during the ongoing annotation process and not before. Researchers should be aware of this and have the required flexibility. A similar issue occurred with regard to words for political professions (e.g., chancellor). This is not provided by the DBoeS, and it was difficult to come up with an exhaustive list and adjust them to the German grammar.[24] On the other side, the bootstrapping approach was messy as it had to be handled manually. For each iteration the author had to manually check the model output. Due to the dataset size this was done using keywords and clustering, but it was not possible to inspect each output sentence individually. However, inspecting only the entities and not the context came with limitations. For names like "Baerbock" it was clear that most likely Annalena Baerbock – a politician – was meant. However, for

[24] The additional entities are listed in the repository at DATA/01_UTIL/Extend.csv.

surnames like "Müller," which are very common in Germany, it was not possible to determine if, for example, Michael Müller, also a politician, was meant. While bootstrapping proved to be very effective, it was limited by these constraints and maybe even more important from a scientific perspective: it limits the reproducibility of this study.

With regard to the practical use of PER, one has to deal with the sometimes messy output of transformer model that is also highlighted by (Balluff et al., 2024). This might not be a problem if PER is used to classify entire documents rather than retrieving individual entities. However, in case of the latter, harmonizing different variations of the same entity can be tedious. One strategy is to aggregate the results, manually search for common patterns, and filter with regular expression. Another option is to use tools like OpenRefine[25] or reconciler[26] that access the WikiData API. Reconciler, for example, can handle "Baerbock," "Annalena_Baerbock," "Baerbock_," and "Annalena_Baerbock_," and makes it possible to link all occurrences to Annalena Baerbock.[27] Nevertheless, this method comes with limitations. For instance, "Außenministerin Baerbock" was not identified. Furthermore, because a country-specific WikiData API has to be chosen, multilingual content from different countries and contexts might complicate this procedure. A possible next step is to develop not only a recognition model for public entities but also a public entity linkage model.

Finally, and with regard to future research, we emphasize that research in communication and political science can benefit from a more fine-grained category system. For example, one could add the remaining types from the DBoeS dataset or add categories like political events, high-ranking employees of the public sector, or NGOs and civil society groups. As our experiments showed, weak supervision, bootstrapping better labels, and fine-tuning on small manually annotated datasets is an efficient way to reach reliable performance. It is likely that a comparable performance can be achieved with even more classes because the main source of confusion was between entities and non-entities and not so much between different entity-classes. This suggests that additional entity-classes might not have a negative effect on the performance. One difficulty, however, could be if new entity-classes frequently co-occur. For instance, adding high-ranking employees of the public sector to the classes could have a negative effect on Authority. We recommend basing an extended taxonomy on a well-grounded schema like the DBoeS. This allows to test and reflect on the best subset of entity classes and can already provide a list of individual instances of the respective entity, that can then be used for weak labeling.

Acknowledgments

We thank Christopher Richter for their assistance in annotating the data and reviewing the literature. We especially appreciate their assistance for the Spanish texts. We further thank Felix Victor Münch, Gregor Wiedemann, and Hadi Asghari for helpful comments and feedback.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The work was supported by the Bundesministerium für Bildung und Forschung [01 IS 20058].

Notes on contributor

Sami Nenno is member of the Public Interest AI research group at Humboldt Institute for Internet and Society and doing his PhD at University of Bremen. In his research he focuses on misinformation and automated tools for fact-checking.

[25] https://openrefine.org/docs
[26] https://jvfe.github.io/reconciler/#quickstart
[27] A notebook with examples can be found at: https://osf.io/4fhze/

ORCID

Sami Nenno http://orcid.org/0009-0009-8150-2558

References

Baden, C., Pipal, C., Schoonvelde, M., & van der Velden, M. A. C. G. (2022). Three gaps in computational text analysis methods for social sciences: A research agenda. *Communication Methods and Measures*, *16*(1), 1–18. https://doi.org/10.1080/19312458.2021.2015574

Balluff, P., Boomgaarden, H. G., & Waldherr, A. (2024). Automatically finding actors in texts: A performance review of multilingual named entity recognition tools. *Communication Methods and Measures*, 1–19. https://doi.org/10.1080/19312458.2024.2324789

Benikova, D., Biemann, C., & Reznicek, M. (2014). NoSta-D named entity annotation for German: Guidelines and dataset. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (pp. 2524–2531). European Language Resources Association (ELRA). https://www.lrec-conf.org/proceedings/lrec2014/pdf/276_Paper.pdf

Chen, P., Xu, H., Zhang, C., & Huang, R. (2022). Crossroads, buildings and neighborhoods: A dataset for fine-grained location recognition. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 3329–3339). https://doi.org/10.18653/v1/2022.naacl-main.243

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8440–8451). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.747

Derczynski, L., Nichols, E., Van Erp, M., & Limsopatham, N. (2017). Results of the WNUT2017 shared task on novel and emerging entity recognition. *Proceedings of the 3rd Workshop on Noisy User-Generated Text* (pp. 140–147). https://doi.org/10.18653/v1/W17-4418

Dettmers, T., Lewis, M., Shleifer, S., & Zettlemoyer, L. (2022). *8-bit optimizers via Block-Wise Quantization* (arXiv:2110.02861). arXiv. https://arxiv.org/abs/2110.02861

Dettmers, T., & Zettlemoyer, L. (2023). The case for 4-bit precision: K-bit inference scaling laws. *Proceedings of the 40th International Conference on Machine Learning* (pp. 7750–7774). https://proceedings.mlr.press/v202/dettmers23a.html

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *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)* (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Jehangir, B., Radhakrishnan, S., & Agarwal, R. (2023). A survey on named entity recognition—datasets, tools, and methodologies. *Natural Language Processing Journal*, *3*, 100017. https://doi.org/10.1016/j.nlp.2023.100017

Kroon, A., Welbers, K., Trilling, D., & van Atteveldt, W. (2023). Advancing automated content analysis for a new era of media effects research: The key role of transfer learning. *Communication Methods and Measures*, 1–21. https://doi.org/10.1080/19312458.2023.2261372

Kumar, A., & Starly, B. (2022). "FabNER": Information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing*, *33*(8), 2393–2407. https://doi.org/10.1007/s10845-021-01807-x

Laurer, M., Atteveldt, W., van Casas, A., & Welbers, K. (2024). Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and BERT-NLI. *Political Analysis*, *32*(1), 84–100. https://doi.org/10.1017/pan.2023.20

Licht, H., & Lind, F. (2023). Going cross-lingual: A guide to multilingual text analysis. *Computational Communication Research*, *5*(2), 1. https://doi.org/10.5117/CCR2023.2.2.LICH

Lison, P., Barnes, J., & Hubin, A. (2021). Skweak: Weak supervision made easy for NLP. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations* (pp. 337–346). https://doi.org/10.18653/v1/2021.acl-demo.40

Lison, P., Barnes, J., Hubin, A., & Touileb, S. (2020). Named entity recognition without labelled data: A weak supervision approach. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 1518–1533). https://doi.org/10.18653/v1/2020.acl-main.139

Liu, K., Fu, Y., Tan, C., Chen, M., Zhang, N., Huang, S., & Gao, S. (2021). Noisy-labeled NER with confidence estimation. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 3437–3445). https://doi.org/10.18653/v1/2021.naacl-main.269

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv. https://doi.org/10.48550/arXiv.1907.11692

Liu, Z., Xu, Y., Yu, T., Dai, W., Ji, Z., Cahyawijaya, S., Madotto, A., & Fung, P. (2021). CrossNER: Evaluating cross-domain named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(15), 13452–13460. https://doi.org/10.1609/aaai.v35i15.17587

Mehrabi, N., Gowda, T., Morstatter, F., Peng, N., & Galstyan, A. (2020). Man is to person as woman is to location: Measuring gender bias in named entity recognition. *Proceedings of the 31st ACM Conference on Hypertext and Social Media* (pp. 231–232). https://doi.org/10.1145/3372923.3404804

Pilny, A., McAninch, K., Slone, A., & Moore, K. (2019). Using supervised machine learning in automated content analysis: An example using relational uncertainty. *Communication Methods and Measures*, *13*(4), 287–304. https://doi.org/10.1080/19312458.2019.1650166

Schmidt, J.-H., Merten, L., & Münch, F. V. (2023). *DBoeS-data [dataset]*. Open Science Framework. https://doi.org/10.17605/OSF.IO/SK6T5

Schweter, S., & Akbik, A. (2021). FLERT: Document-level features for named entity recognition. *arXiv :2011.06993 [Cs]*. http://arxiv.org/abs/2011.06993

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. *Proceedings of the 34th International Conference on Machine Learning* - (Vol. *70*. pp. 3319–3328).

Tänzer, M., Ruder, S., & Rei, M. (2022). Memorisation versus generalisation in Pre-trained language models. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 7564–7578). https://doi.org/10.18653/v1/2022.acl-long.521

Tjong Kim Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003* (pp. 142–147). https://doi.org/10.48550/arXiv.cs/0306050

Tolochko, P., Balluff, P., Bernhard, J., Galyga, S., Lebernegg, N. S., & Boomgaarden, H. G. (2024). What's in a name? The effect of named entities on topic modelling interpretability. *Communication Methods and Measures*, 1–22. https://doi.org/10.1080/19312458.2024.2302120.

Trilling, D., & Jonkman, J. G. F. (2018). Scaling up content analysis. *Communication Methods and Measures*, *12*(2–3), 158–174. https://doi.org/10.1080/19312458.2018.1447655

Ushio, A., Barbieri, F., Sousa, V., Neves, L., & Camacho-Collados, J. (2022). Named entity recognition in Twitter: A dataset and analysis on short-term temporal shifts. In Y. He, H. Ji, S. Li, Y. Liu, & C.-H. Chang (Eds.), *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 309–319). Association for Computational Linguistics. https://doi.org/10.48550/arXiv.2210.03797

Van Aelst, P., Sheafer, T., & Stanyer, J. (2012). The personalization of mediated political communication: A review of concepts, operationalizations and key findings. *Journalism*, *13*(2), 203–220. https://doi.org/10.1177/1464884911427802

van Atteveldt, W., & Peng, T.-Q. (2018). When communication meets computation: Opportunities, challenges, and pitfalls in computational communication science. *Communication Methods and Measures*, *12*(2–3), 81–92. https://doi.org/10.1080/19312458.2018.1458084

van Atteveldt, W., van der Velden, M. A. C. G., & Boukes, M. (2021). The validity of sentiment analysis: Comparing manual annotation, crowd-coding, dictionary approaches, and machine learning algorithms. *Communication Methods and Measures*, *15*(2), 121–140. https://doi.org/10.1080/19312458.2020.1869198

Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, Z., Huang, F., & Tu, K. (2021). Improving named entity recognition by external context retrieving and cooperative learning. *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)* (pp. 1800–1812). https://doi.org/10.18653/v1/2021.acl-long.142

Weischedel, R., Hovy, E., Marcus, M., Palmer, M., Belvin, R., Pradhan, S., Ramshaw, L., & Xue, N. (2013). OntoNotes: A large training corpus for enhanced processing [dataset]. https://doi.org/10.35111/xmhb-2b84

Yu, J., Bohnet, B., & Poesio, M. (2020). Named entity recognition as dependency parsing. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 6470–6476). https://doi.org/10.18653/v1/2020.acl-main.577

Yu, Y., Zuo, S., Jiang, H., Ren, W., Zhao, N. A. I., & Zhang, C. (2021). Fine-tuning Pre-trained language Model with weak supervision: A contrastive-regularized self-training approach. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, & Y. Zhou (Eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1063–1077). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.84

Zhu, D., Shen, X., Mosbach, M., Stephan, A., & Klakow, D. (2023). Weaker than you think: A critical look at weakly supervised learning. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 14229–14253). https://doi.org/10.18653/v1/2023.acl-long.796

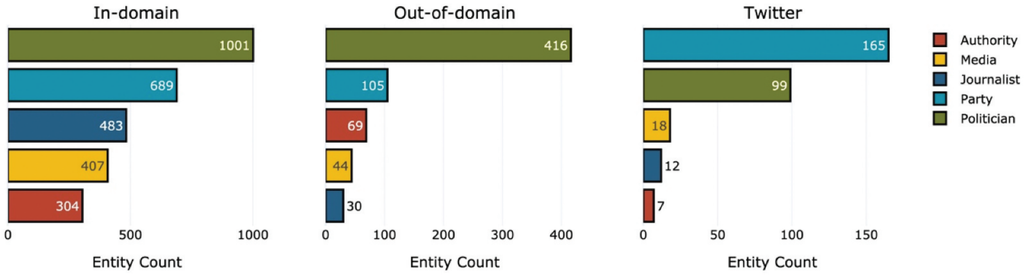
Appendix

Table A1. Class-specific (weighted) F_1, recall, and precision of baseline of weak annotators and XLM-R fine-tuned on weak data only (FT_w), clean data only (FT_c), and continuous fine-tuning on weak and clean data (CFT) over all iterations.

		F_1					Recall					Precision				
		ID	OD	Tw	En	Sp	ID	OD	Tw	En	Sp	ID	OD	Tw	En	Sp
CFT Base	Politician	**.83**	.80	.82	**.70**	.69	**.80**	.79	.84	.67	.69	**.86**	.80	.79	.74	.70
	Party	.82	**.81**	**.86**	.48	.73	**.80**	**.80**	**.88**	.42	.69	.85	**.82**	**.84**	.57	.79
	Authority	.70	.64	.40	.25	.39	.67	.71	.52	.41	.39	.73	.59	.34	.18	.39
	Media	.79	.68	.72	.27	.85	.79	.66	.68	.23	.86	.80	.70	.77	.33	.84
	Journalist	.79	.56	.55	.00	**.86**	.75	.48	.83	.00	**.90**	.84	.68	.41	.00	**.86**
CFT Large	Politician	**.85**	**.83**	.82	**.75**	.67	**.83**	**.85**	.83	.73	.69	**.88**	.82	.81	**.79**	.66
	Party	.82	**.83**	**.88**	.62	.73	.80	.81	.88	**.77**	.73	.84	**.84**	**.88**	.51	.73
	Authority	.74	.62	.36	.28	.40	.72	.80	**1.0**	.59	.47	.78	.52	.22	.19	.36
	Media	.80	.60	.73	.29	**.86**	.77	.68	.65	.26	**.81**	.83	.55	.84	.33	.92
	Journalist	.82	.63	.53	.00	.83	.80	.52	.73	.00	.72	.84	.82	.41	.00	**1.0**
Base-line	Politician	.48	.31	.26	.00	.00	.37	.19	.16	.00	.00	.67	.80	.76	.00	.00
	Party	**.68**	**.68**	**.79**	.00	.00	**.58**	**.58**	**.72**	.00	.00	**.82**	**.82**	.89	.00	.00
	Authority	.50	.36	.22	.00	.00	.39	.24	.14	.00	.00	.67	.73	.50	.00	.00
	Media	.25	.15	.36	.00	.00	.15	.09	.22	.00	.00	.80	.50	**1.0**	.00	.00
	Journalist	.22	.00	.00	.00	.00	.13	.00	.00	.00	.00	.66	.00	.00	.00	.00
FT_c Base	Politician	**.79**	.73	.56	**.56**	.59	**.74**	.71	.67	**.52**	.53	**.87**	**.85**	.55	**.72**	.73
	Party	.76	**.77**	**.84**	.34	.64	**.74**	**.76**	.83	.26	.55	.80	.81	**.87**	.50	.79
	Authority	.56	.43	.70	.24	.30	.62	.59	**1.0**	.37	.28	.73	.40	.60	.18	.31
	Media	.73	.63	.57	.18	**.76**	**.74**	.63	.54	.14	**.64**	.79	.70	.67	.33	**.97**
	Journalist	.67	.49	.23	.00	.18	.65	.38	.14	.00	.15	.81	.67	.43	.00	.73
FT_c Large	Politician	**.81**	**.80**	.56	**.60**	.66	**.78**	**.79**	.63	**.60**	.62	**.86**	**.85**	.51	**.70**	.79
	Party	.76	.79	**.82**	.47	.72	.73	.77	**.80**	.46	.68	.81	.84	**.86**	.48	.80
	Authority	.61	.46	.14	.26	.37	.57	.56	**.80**	.32	.36	.75	.45	.11	.20	.35
	Media	.74	.51	.71	.28	**.87**	.71	.54	.65	.25	**.86**	.79	.56	.80	.33	.89
	Journalist	.72	.48	.21	.00	.63	.68	.38	.20	.00	.45	.84	.77	.37	.0	**1.0**

Table A2. Statistics of model performance and attribution scores on datasets with different levels of label-noise.

Shuffled labels (%)	Max F_1	Memorized predictions (%)	Avg. attribution score
0	0.9	39.05	0.49
70	0.67	40.55	0.51
80	0.67	43.39	0.52
90	0.68	42.48	0.51
100	0.69	40.79	0.50



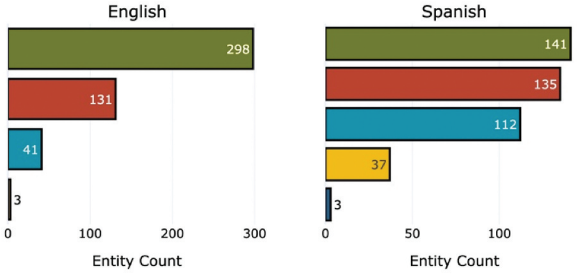


Figure A1. Entity distribution for manually annotated datasets.

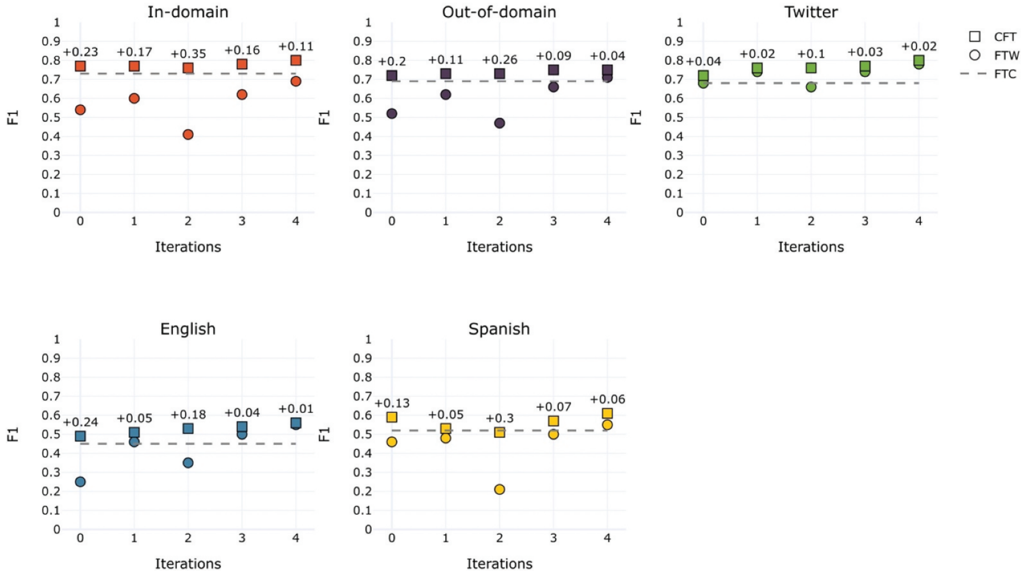


Figure A2. F_1-scores of fine-tuning XLM-R base on weak data only (FT_w), clean data only (FT_c), and continuous fine-tuning on weak and clean data (CFT) over all iterations. Numbers indicate the performance increase from FT_w to CFT.

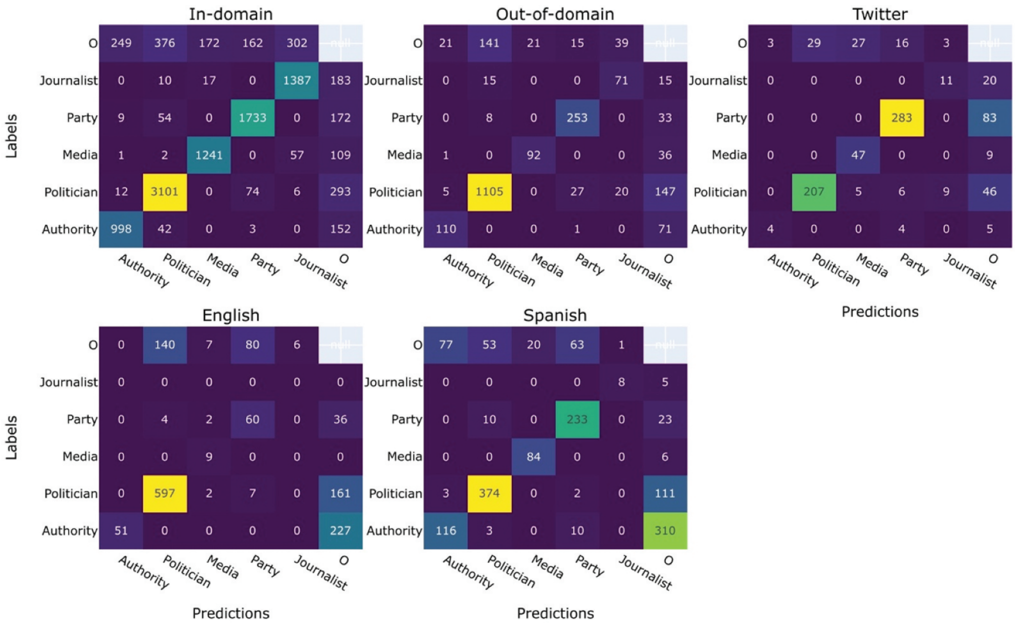


Figure A3. Confusion matrices for CFT Base in iteration 4. “O” denotes tokens that did not fall into any of the entity-categories. We left the right upper corner of each matrix empty for better readability.

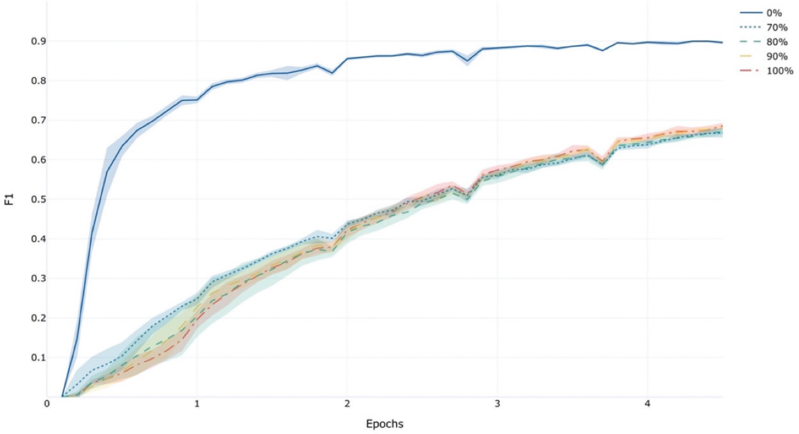


Figure A4. Training curve of XLM-R base fine-tuned on datasets with different levels of label-noise.