Evaluation of Data Augmentation for Named Entity Recognition in the German Legal Domain

Robin Erd[1], Leila Feddoul[1,*], Clara Lachenmaier[2] and Marianne Jana Mauch[1]

[1]*Heinz Nixdorf Chair for Distributed Information Systems, Friedrich Schiller University Jena, Jena, Germany*
[2]*Computatitional Linguistics, Bielefeld University, Bielefeld, Germany*

Abstract

One of the techniques to solve Natural Language Processing tasks is supervised learning, which requires large labeled datasets for model training. Such datasets are usually unavailable for specific domains or languages other than English. Creating them manually is a time-consuming task. This paper aims to explore methods to artificially expand small datasets in the German legal domain. We tested three Data Augmentation approaches on differently sized fragments of the German Legal Entity Recognition dataset: Synonym replacement, mention replacement, and back translation. We evaluated the effect of training on the augmented data with a bidirectional Long Short-Term Memory Network with a Conditional Random Field layer and a Transformer-based model. It appears that synonym replacement and mention replacement yield similarly positive results, while the latter is less time-consuming. Performing back translation turns out as challenging using legal texts.

Keywords

Named Entity Recognition, Data Augmentation, German Legal Domain

1. Introduction

The project Canaréno[1] aims to support the analysis of German legal norms, which is done as a first step in the creation of digital administrative processes. In general, German public authorities need to follow a specific process in order to deliver an administrative service (e.g., vehicle registration) for citizens or companies. This process is not arbitrary created, but it is typically based on legal bases (e.g., laws, ordinances, etc.). Thus, the very first step for the modeling of administrative processes is to gather relevant legal bases and to analyze them in a subsequent step. The purpose of this analysis is to identify indications in the text about possible process elements (e.g., process steps, participants, etc.). This is carried out manually in either an implicit, or an explicit way by highlighting relevant words or sentences belonging to specific categories such as *process main actor* or *process contributor*. This task is not only time- and personnel-consuming, but also requires a certain expertise. Therefore, the initial goal is

Joint Proceedings of ISWC2022 Workshops: the International Workshop on Artificial Intelligence Technologies for Le Documents (AI4LEGAL) and the International Workshop on Knowledge Graph Summarization (KGSum) (2022)

[*] Corresponding author.

robin.erd@uni-jena.de (R. Erd); leila.feddoul@uni-jena.de (L. Feddoul); clara.lachenmaier@uni-bielefeld.de (C. Lachenmaier); marianne.mauch@uni-jena.de (M. J. Mauch)

0000-0001-5340-8163 (R. Erd); 0000-0001-8896-8208 (L. Feddoul); 0000-0002-9207-3420 (C. Lachenmaier); 000-0003-1478-1867 (M. J. Mauch)

[1] http://www.opendva.uni-jena.de/

to support this legal analysis by performing automatic suggestions about relevant objects. In this context, Named Entity Recognition (NER) techniques are investigated. NER aims to detect and classify Named Entities (NEs), e.g., persons in unstructured text. It allows machines to better understand the contained information and serves as an initial step for performing more complex tasks (e.g., question answering). In our context, it will be used as a basis for further processing to identify possible process steps and their interaction with other process elements. However, common NE classes are often reflecting generic concepts. As soon as texts cover specific domains that deal with particular phenomena, general NER classes do not suffice to depict all concepts of the certain niche. Recent datasets used special domain-specific tags for legal texts, such as *lawyer*, *legal norm*, or *court* [1]. Nevertheless, the dataset is not covering the specific properties related to administrative processes. To the best of our knowledge, no labeled dataset exists using those specific tags.

NER is often solved by training supervised machine learning models, which learn complex features, when they are provided with sufficient labeled data. The process of manual data labeling could involve domain experts and is time-consuming. In this context, we want to investigate techniques for Data Augmentation (DA) using another dataset similar in nature to our target data. Small datasets can be enlarged by generating new training data automatically using DA. Some of the methods for DA in the field of Natural Language Processing (NLP) and specifically NER are, among others: synonym replacement [2, 3, 4, 5], mention replacement[2] [6, 5, 7], random deletion [3], random insertion [3], random swap [3, 5], noising techniques [3], TF-IDF based word replacement [8], back translation [9], and generative approaches[10, 11, 12].

There are works considering various aspects of different DA techniques, but to the best of our knowledge: (1) none uses data from the legal domain and all of them mainly consider English data, (2) none compares different sources that can be used with the synonym replacement, and (3) other implementations of back translation only perform segment-wise back translation while excluding entities, limiting the degree of change that might be achieved, or employ a NER model to re-annotate the back translated sentences.

Our goal is to evaluate and compare DA techniques for the NER task, explicitly focusing on the German legal domain and thus using the German Legal Entity Recognition (LER) dataset [1]. The key contributions of this paper are:

1. A workflow for the generation and augmentation of different dataset fractions using three different DA techniques along with three different synonym sources.
2. A back translation method that (1) does not rely on pre-trained models for translation or re-annotation, and (2) translates the whole sentence, including entities, and thereby enriches the mentions space.
3. Evaluation and comparison of the effectiveness of the selected DA approaches using two deep learning models on a German legal dataset.

The source code for data generation and evaluation is publicly available [13, 14] under an MIT License. Generated datasets [15] and evaluation results [16] are published on Zenodo.

[2] Note that in NER, the terms "mention" and "entity" can be used interchangeably.

2. Related Work

Named Entity Recognition. With most research in NLP and specifically NER being conducted in the English language, few works exist regarding NER in the German legal domain. Glaser et al. [17] tested three techniques for extracting entities from German legal contracts: GermaNER, DBpedia Spotlight [18], and templated NER. GermaNER and DBpedia Spotlight achieved an F1-score of 0.80 and 0.87 respectively, while templated NER was tested on a smaller dataset and achieved an F1-score of 0.92. Leitner et al. [19] evaluated different Bidirectional Long Short-Term Memory (BiLSTM) networks with a Conditional Random Field (CRF) layer for NER on the German LER dataset. The best performance was achieved using two BiLSTM-CRF models with character embeddings with an F1-score of 0.9546. More recently, Zöllner et al. [20] compared different pre-training techniques and a modified fine-tuning process for small Bidirectional Encoder Representations from Transformers (BERT) [21] models and also used the LER dataset for evaluation, achieving an F1-score of 0.9488.

Data Augmentation. *Replacement-based Techniques.* The replacement of words was one of the first techniques to be employed for DA. Zhang et al. [2] applied WordNet-based [22] synonym replacement to eight text classification datasets. Wang et al. [23] applied Word2vec-based synonym replacement to a newly created Twitter dataset used for topic classification. Wu et al. [24] randomly replaced one to two words per sentence with a [MASK] token, which was then to be filled by a label-conditioned contextual language model. Liu et al. [7] randomly replaced mentions in the training data with mentions from a manually created dictionary containing mentions not part of the training data.

Combined Techniques. Wei et al. [3] presented four techniques for use with text classification data now known as easy-data-augmentation (EDA) techniques, namely synonym replacement, random insertion, random swap, and random deletion. Kang et al. [25] extended these, adding an external knowledge-based system and modifying them to work with NER tasks. Shim et al. [4] and Issifu et al. [26] adapted the modified EDA techniques. Dai et al. [5] applied label-wise token replacement, synonym replacement, mention replacement, and shuffle within segments techniques to the MaSciP (materials science) and i2b2-2010 (biomedical) NER datasets. Yaseen et al. [9] used the same techniques as Dai et al., but applied them to the MaSciP and Species-800 datasets.

Back Translation. Xie et al. [8] applied back translation to a topic classification dataset, Luque et al. [27] used back translation to augment a sentiment analysis dataset. While all these applications are sentence-level tasks, there has recently been an effort to apply back translation to sequence-labeling data required by, e.g., the NER task. Yaseen et al. [9] applied segment-wise back translation to the MaSciP and Species-800 datasets, back translating only the context, excluding the entities. They achieved an increase in F1-score of 0.0645 and 0.0148, respectively. Sabty et al. [28] applied back translation to Arabic-English code-switching[3] NER data but did not improve performance. Their task differs from our task as they also had to preserve the code-switching property of the sentences. They re-annotated the back translated sentences using a NER model, tested using trained models for translation instead of Google Translate, and tried more than one pivotal language.

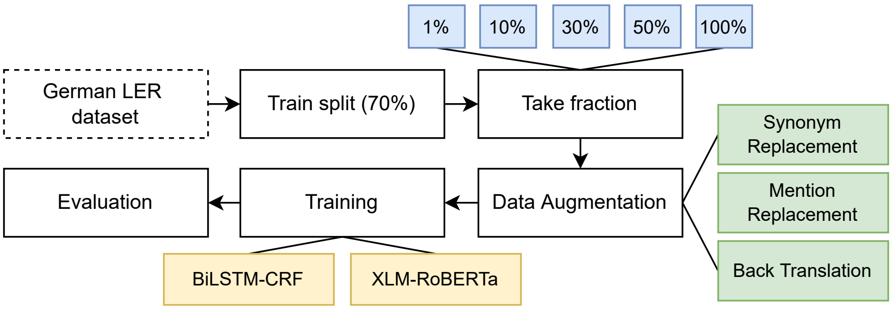
[3] Code-switching refers to text containing more than one language in the same sentence.

While previously mentioned works try to evaluate different DA approaches, they focus on general domains and mostly consider English datasets. Furthermore, we do not find any comparison of different synonym replacement sources. Considering back translation, the mentioned works either only back translate the context of entities or use a model to perform the translation and re-annotation.

3. Approach

Figure 1 depicts our proposed workflow. We establish a baseline by considering the train split of the chosen dataset, taking different fractions (%): {1, 10, 30, 50, and 100}, training two models with the non-augmented data, and evaluating the performance of each model on the test split. To evaluate the impact of DA techniques on the model performance, we apply the selected DA techniques to these training split fractions, train on the generated augmented fractions, and evaluate the impact of DA. Table 1 illustrates examples for each technique.

Figure 1: Overview of the proposed approach.



We implement and evaluate the following three DA techniques, each of which attempts to create a modified copy of each sentence in the training dataset. A modified copy of a sentence can only be created if all technique-specific conditions are met. This leads to varying numbers of augmented sentences between the applied DA techniques. In addition, only sentences whose tokenization is reproducible can be augmented. One could also augment the dataset iteratively, generating multiple augmented sentences for each original sentence, but in that case a mechanism should be applied to avoid duplicates and ensure sufficient degrees of variation. We have applied just one augmentation iteration round. Generated sentences are appended to the original training dataset.

Synonym Replacement. We substitute a percentage of not-tagged, qualified[4] tokens in the sentence with a replacement similar in meaning. We compare three different external sources for replacements: OpenThesaurus [29], fastText embeddings [30], and the contextual language model XLM-RoBERTa [31]. The augmentation of a sentence only succeeds if the selected replacement source provides a replacement for the selected tokens and at least one token qualifies for replacement (e.g., numbers are not qualified). Furthermore, the selected replacement percentage has to amount to at least one token.

[4] We filter replacement candidates with a regular expression to avoid replacing or inserting, e.g., punctuation.

Table 1
Example of augmented sentences with synonym replacement (SR), mention replacement (MR), and back translation (BT) with changes highlighted in magenta.

None	Alex	is	going	to	Los	Angeles	in	California	.
SR	Alex	was	walking	towards	Los	Angeles	around	California	.
MR	Chloe	is	going	to	Mexico	in	United	Kingdom	.
BT	Alex	is	going	to	Los	Angeles	,	California	.

Mention Replacement. For each mention in the sentence, we replace it with a random mention of the same class from the original training set. Only sentences containing mentions can be augmented using this technique.

Back Translation. We first extract all mentions and their class from the sentence. We then back translate the complete sentence as a plain string and the extracted mentions separately using the `BackTranslation`[5] python package, which depends on external services to provide translations. We then map the extracted mentions back to the back translated sentence based on their token sequence[6]. As pivotal language, we decided to use English. With this process, we essentially preserve the original labels and adapt them to the new sentence, adding new mention variants to the dataset and foregoing the need to use a NER model to perform the re-annotation.

4. Experiments

4.1. Experimental setup

All experiments were performed on AlmaLinux 8.3 using Python 3.9.12. Training and evaluation are run on a single NVIDIA A100 GPU.

Dataset. We evaluate the DA methods on the German LER dataset, containing $\approx 67,000$ sentences with over 2 million tokens classified into 19 fine-grained semantic classes[7]. As the train/dev/test splits are not provided, we split the data ourselves to $70/15/15$ splits. Consequently, our training split contains $46,706$ sentences. We work with IOB2 [32], the tagging scheme in which the data is provided. When working with the data during augmentation, the tokenization of the original sentence should be reproduced. Therefore we use the `SoMaJo` tokenizer [33] used by Leitner et al [1].

NER Models. To evaluate the effect of the DA techniques, we choose two models. One is BiLSTM-CRF, implemented using the `FLAIR` framework [34]. Following the recommendations of Akbik et al. [35], we use it with stacked German fastText and German forward and backward `FLAIR` embeddings, train the model using Stochastic Gradient Descent without momentum, clip gradients at 5, and anneal the learning rate against the micro F1-score on the dev split,

[5] https://pypi.org/project/BackTranslation/, accessed on 03.08.2022

[6] This is only possible if a sentence does not contain a token sequence multiple times with different label sequences each. If mapping the extracted mentions to the new sentence fails, the augmentation of this particular original sentence is canceled.

[7] Leitner et al. found that some tags, such as *street*, *landscape*, *brand* and *regulation* are more difficult to predict than e.g., *judge*, *law* and *court*.

halving it if the score does not increase for 5 consecutive epochs. We use a learning rate of 0.05, a mini-batch size of 32, apply variational dropout and train the model for 150 epochs but stop earlier if the learning rate falls below 0.0001.

The other model is a Transformer-based model, implemented using the FLERT extension [36] of the FLAIR framework. We chose the XLM-RoBERTa Transformer model (XLM-R) over models trained specifically for the German language as preliminary studies showed that it achieves better results on the LER dataset than, e.g., GELECTRA [37]. We fine-tune it for 10 epochs using the AdamW optimizer with a mini-batch size of 1. The learning rate increases from 0 to $5e-6$ during the warm-up phase and then linearly decreases, reaching 0 by the end of the training.

4.2. Results

Table 2 provides the baseline results on the test set before applying the DA techniques as well as the results after applying synonym replacement. We use the micro F1-score to evaluate model performance. Baseline results show that in very low-data settings, BiLSTM-CRF outperforms XLM-R. For all other dataset fractions, XLM-R outperforms BiLSTM-CRF. Note that a relatively good performance is achieved with only 10% and 30% of the original dataset.

Synonym Replacement. Depending on the selected configuration, synonym replacement is the most expensive technique, taking up to 12 seconds per sentence when using the contextual language model as source in combination with a replacement percentage of 60%. The augmentation of the German LER dataset consequently took between 5 and 155 hours, boosting the dataset size by up to 87.3% when applied with a replacement percentage of 60% and using fastText or the contextual language model as source. Our replacement percentages of 20%, 40% and 60% of eligible tokens result in around 10.8%, 22.7%, and 34.6% of total tokens being replaced.

OpenThesaurus. Using OpenThesaurus as source, we notice that a higher replacement percentage improves XLM-R performance, while for BiLSTM-CRF, it does not. XLM-R trained on the larger datasets benefits marginally from applying DA, while for BiLSTM-CRF, we get a mixed picture.

fastText. With fastText embeddings, a higher replacement percentage improves BiLSTM-CRF performance, while for XLM-R , it does not. Additionally, we notice that the performance of the XLM-R model after training on the larger datasets is impacted very slightly by DA. In contrast, BiLSTM-CRF shows improvements for the 50%-dataset.

Contextual Language Model. For the 1%, 10%, and 30% datasets, XLM-R benefits more than BiLSTM-CRF. In contrast to other sources, we notice that a higher replacement percentage does not increase but reduces the augmentation's positive impact across all dataset and model combinations. The augmentation affects the performance on the 100%-dataset only marginally.

Overall. Figure 2 shows the average relative improvement in micro F1-score across all datasets achieved by applying synonym replacement. We notice that XLM-R benefits more from DA than BiLSTM-CRF, with the contextual language model as source yielding the greatest improvement. Applying synonym replacement leads to improvements in most cases. We deduce that the contextual language model is best used with a low replacement percentage.

Mention Replacement. Mention replacement is the least expensive technique, with the

Table 2
Evaluation results after training on data augmented with synonym replacement (SR) in terms of micro F1-score with either OpenThesaurus (THE), fastText embeddings (FTX), or a contextual language model (CLM) as replacement source and different replacement percentages.

		BiLSTM-CRF				XLM-R			
Dataset	**Source**	**Base**	**20%**	**40%**	**60%**	**Base**	**20%**	**40%**	**60%**
1%	THE	0.6994	-0.0035	**+0.0108**	+0.0096	0.6089	+0.0081	-0.0028	**+0.0322**
10%	THE	0.8941	**+0.0130**	+0.0073	+0.0042	0.9130	+0.0079	+0.0054	**+0.0104**
30%	THE	0.9346	+0.0010	+0.0012	**+0.0014**	0.9416	+0.0061	**+0.0097**	+0.0084
50%	THE	0.9430	**+0.0059**	+0.0033	+0.0040	0.9559	-0.0007	+0.0021	**+0.0031**
100%	THE	0.9572	+0.0023	**+0.0031**	+0.0013	0.9661	+0.0007	**+0.0010**	+0.0004
Avg.			+0.0037	**+0.0051**	+0.0041		+0.0044	+0.0031	**+0.0109**
1%	FTX	0.6994	-0.0180	-0.0030	**+0.0072**	0.6089	**+0.0296**	+0.0224	+0.0268
10%	FTX	0.8941	+0.0047	+0.0073	**+0.0079**	0.9130	+0.0033	**+0.0055**	+0.0051
30%	FTX	0.9346	+0.0009	+0.0009	**+0.0023**	0.9416	**+0.0089**	+0.0063	+0.0075
50%	FTX	0.9430	**+0.0053**	+0.0048	+0.0049	0.9559	+0.0012	+0.0001	**+0.0022**
100%	FTX	0.9572	+0.0005	**+0.0025**	+0.0023	0.9661	+0.0011	**+0.0014**	+0.0002
Avg.			-0.0013	+0.0025	**+0.0049**		**+0.0088**	+0.0071	+0.0084
1%	CLM	0.6994	**+0.0180**	-0.0039	+0.0007	0.6089	**+0.0943**	+0.0770	+0.0530
10%	CLM	0.8941	+0.0015	**+0.0089**	-0.0018	0.9130	**+0.0074**	+0.0018	+0.0013
30%	CLM	0.9346	+0.0002	**+0.0008**	+0.0002	0.9416	+0.0045	**+0.0047**	+0.0028
50%	CLM	0.9430	**+0.0047**	**+0.0047**	+0.0035	0.9559	**+0.0020**	-0.0004	-0.0005
100%	CLM	0.9572	**+0.0006**	-0.0005	-0.0018	0.9661	-0.0015	**-0.0003**	-0.0013
Avg.			**+0.0050**	+0.0020	+0.0002		**+0.0213**	+0.0166	+0.0111

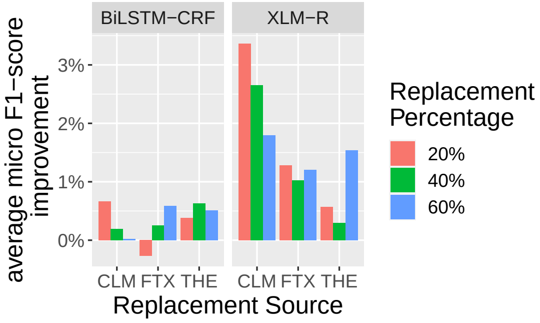
Table 3
Evaluation results after training on data augmented with mention replacement (MR) or back translation (BT) in terms of micro F1-score.

	BiLSTM-CRF			XLM-R		
Dataset	**Base**	**MR**	**BT**	**Base**	**MR**	**BT**
1%	0.6994	**+0.0222**	+0.0065	0.6089	**+0.0772**	-0.0123
10%	0.8941	**+0.0053**	-0.0040	0.9130	**+0.0103**	-0.0025
30%	0.9346	**+0.0032**	+0.0006	0.9416	**+0.0064**	+0.0037
50%	0.9430	**+0.0061**	+0.0063	0.9559	**+0.0033**	-0.0008
100%	0.9572	**+0.0007**	-0.0003	0.9661	**-0.0003**	-0.0004
Avg.		**+0.0075**	+0.0018		**+0.0194**	-0.0025

augmentation taking only 0.011 seconds per sentence. It increased the dataset size by 37.854%. Table 3 lists the results achieved after training both models on the augmented datasets. We notice that the improvements for all datasets larger than the 10%-dataset are minor. The maximum relative improvement is achieved with the 1%-dataset for both models. The average change in micro F1-score across all datasets is +0.0075 and +0.0194 for BiLSTM-CRF and XLM-R. We also evaluated, although without notable results, the effect of applying both mention and synonym replacement combined.

Back Translation. By applying back translation, we were able to increase the dataset size by 63.24%, boosting the total number of annotated entities by 17.52%. However, we do not register a significant impact on the performance regarding the micro F1-score of either BiLSTM-CRF or

Figure 2: Average micro F1-score improvements across all datasets achieved by applying synonym replacement by source, model, and replacement percentage.



XLM-R. The average change in micro F1-score across all datasets is $+0.0018$ and -0.0025 for BiLSTM-CRF and XLM-R.

5. Conclusion and Future Work

We implemented three different DA techniques for use with NER training data, evaluated them on data from the German legal domain, and compared different German replacement sources and percentages for synonym replacement. We believe that the proposed implementation of back translation is unique in its ability to back translate entire sentences while preserving their labels. Our workflow included two models and five different fractions of the full dataset.

We found that DA can be beneficial when working with small datasets, such as the 1% dataset, containing only 468 sentences. Considering that synonym and mention replacement deliver comparable improvements, the latter is the most efficient. For synonym replacement, the contextual language model is the most effective source. Back translation is challenged by the long and nested sentences, occasional ambiguities, and frequently occurring legal concepts that do not exist in the country's legal system of the used pivotal language. Back translation achieved a maximum improvement of $+0.0065$ using BiLSTM-CRF with the 1%-dataset. Mention replacement achieved a maximum improvement of $+0.0772$ using XLM-R with the 1%-dataset. Synonym replacement achieved a maximum improvement of $+0.0943$ using XLM-R with the 1%-dataset, a replacement percentage of 20% and the contextual language model as source.

Future work could focus on improving the proposed back translation technique by, e.g., adding more flexibility to the re-annotation process. Mention replacement could be extended to get the replacements from, e.g., a knowledge base, to introduce new entities. Synonym replacement could benefit from a mechanism that prevents the replacements from being too similar to the original token. In the context of the Canaréno project, this evaluation gives us more insights about which techniques are better suited to augment our small manually annotated dataset, before applying and evaluating different NER models.

Acknowledgments

The project Canaréno was funded by the Federal Ministry of the Interior and Community and the Free State of Thuringia. We also thank Prof. Dr. Birgitta König-Ries and Pr. Dr. Sina Zarrieß for the guidance and feedback.

References

[1] E. Leitner, G. Rehm, J. Moreno-Schneider, A Dataset of German Legal Documents for Named Entity Recognition, in: LREC 2020, European Language Resources Association, 2020, pp. 4478–4485. URL: https://aclanthology.org/2020.lrec-1.551.

[2] X. Zhang, J. J. Zhao, Y. LeCun, Character-level Convolutional Networks for Text Classification, in: NIPS 2015, 2015, pp. 649–657. URL: https://dl.acm.org/doi/10.5555/2969239.2969312.

[3] J. Wei, K. Zou, EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks, in: EMNLP@IJCNLP 2019, Association for Computational Linguistics, 2019, pp. 6382–6388. doi:10.18653/v1/D19-1670.

[4] H. Shim, S. Luca, D. Lowet, B. Vanrumste, Data Augmentation and Semi-Supervised Learning for Deep Neural Networks-Based Text Classifier, Association for Computing Machinery, 2020, pp. 1119–1126. doi:10.1145/3341105.3373992.

[5] X. Dai, H. Adel, An Analysis of Simple Data Augmentation for Named Entity Recognition, in: COLING 2020, International Committee on Computational Linguistics, 2020, pp. 3861–3867. doi:10.18653/v1/2020.coling-main.343.

[6] J. Raiman, J. Miller, Globally Normalized Reader, in: EMNLP 2017, Association for Computational Linguistics, 2017, pp. 1059–1069. doi:10.18653/v1/D17-1111.

[7] Q. Liu, P. Li, W. Lu, Q. Cheng, Long-tail Dataset Entity Recognition based on Data Augmentation, in: EEKE@JCDL 2020, CEUR-WS.org, 2020, pp. 79–80. URL: http://ceur-ws.org/Vol-2658/paper10.pdf.

[8] Q. Xie, Z. Dai, E. H. Hovy, T. Luong, Q. Le, Unsupervised Data Augmentation for Consistency Training, in: NIPS 2020, 2020. URL: https://dl.acm.org/doi/10.5555/3495724.3496249.

[9] U. Yaseen, S. Langer, Data Augmentation for Low-Resource Named Entity Recognition Using Backtranslation (2021). doi:10.48550/ARXIV.2108.11703.

[10] A. Keraghel, K. Benabdeslem, B. Canita, Data augmentation process to improve deep learning-based NER task in the automotive industry field, in: IJCNN 2020, 2020, pp. 1–8. doi:10.1109/IJCNN48605.2020.9207241.

[11] R. Zhou, X. Li, R. He, L. Bing, E. Cambria, L. Si, C. Miao, MELM: Data Augmentation with Masked Entity Language Modeling for Low-Resource NER (2022) 2251–2262. doi:10.18653/v1/2022.acl-long.160.

[12] R. Zhang, Y. Yu, C. Zhang, SeqMix: Augmenting Active Sequence Labeling via Sequence Mixup, in: EMNLP 2020, Association for Computational Linguistics, 2020, pp. 8566–8579. doi:10.18653/v1/2020.emnlp-main.691.

[13] R. Erd, L. Feddoul, fusion-jena/data-augmentation-ner-legal, 2022. URL: https://github.com/fusion-jena/data-augmentation-ner-legal.

[14] R. Erd, L. Feddoul, fusion-jena/data-augmentation-ner-legal v1.0.1, 2022. doi:10.5281/zenodo.6992392.

[15] R. Erd, L. Feddoul, C. Lachenmaier, M. J. Mauch, data-augmentation-ner-datasets, 2022. doi:10.5281/zenodo.6956603.

[16] R. Erd, L. Feddoul, C. Lachenmaier, M. J. Mauch, data-augmentation-ner-results, 2022. doi:10.5281/zenodo.6956508.

[17] I. Glaser, B. Waltl, F. Matthes, Named entity recognition, extraction, and linking in German legal contracts, in: IRIS: Internationales Rechtsinformatik Symposium, 2018, pp. 325–334.

[18] P. N. Mendes, M. Jakob, A. García-Silva, C. Bizer, DBpedia spotlight: shedding light on the web of documents, in: I-SEMANTICS 2011, ACM International Conference Proceeding Series, ACM, 2011, pp. 1–8. doi:10.1145/2063518.2063519.

[19] E. Leitner, G. Rehm, J. Moreno-Schneider, Fine-Grained Named Entity Recognition in Legal Documents, in: SEMANTiCS 2019, Springer International Publishing, 2019, pp. 272–287. doi:10.1007/978-3-030-33220-4_20.

[20] J. Zöllner, K. Sperfeld, C. Wick, R. Labahn, Optimizing Small BERTs Trained for German NER, Inf. 12 (2021) 443. doi:10.3390/info12110443.

[21] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: NAACL-HLT 2019, Association for Computational Linguistics, 2019, pp. 4171–4186. doi:10.18653/v1/n19-1423.

[22] C. Fellbaum, WordNet: An Electronic Lexical Database, The MIT Press, 1998. doi:10.7551/mitpress/7287.001.0001.

[23] W. Y. Wang, D. Yang, That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets, in: EMNLP 2015, Association for Computational Linguistics, 2015, pp. 2557–2563. doi:10.18653/v1/D15-1306.

[24] X. Wu, S. Lv, L. Zang, J. Han, S. Hu, Conditional BERT Contextual Augmentation, in: ICCS 2019, Springer International Publishing, 2019, pp. 84–95. doi:10.1007/978-3-030-22747-0_7.

[25] T. Kang, A. Perotte, Y. Tang, C. Ta, C. Weng, UMLS-based data augmentation for natural language processing of clinical research literature, Journal of the American Medical Informatics Association 28 (2020) 812–823. doi:10.1093/jamia/ocaa309.

[26] A. M. Issifu, M. C. Ganiz, A Simple Data Augmentation Method to Improve the Performance of Named Entity Recognition Models in Medical Domain, in: 2021 6th International Conference on Computer Science and Engineering (UBMK), 2021, pp. 763–768. doi:10.1109/UBMK52708.2021.9558986.

[27] F. M. Luque, Atalaya at TASS 2019: Data Augmentation and Robust Embeddings for Sentiment Analysis, in: IberLEF@SEPLN 2019, CEUR-WS.org, 2019, pp. 561–570. URL: http://ceur-ws.org/Vol-2421/TASS_paper_1.pdf.

[28] C. Sabty, I. Omar, F. Wasfalla, M. Islam, S. Abdennadher, Data Augmentation Techniques on Arabic Data for Named Entity Recognition, Procedia Computer Science 189 (2021) 292–299. doi:10.1016/j.procs.2021.05.092.

[29] D. Naber, OpenThesaurus: ein offenes deutsches Wortnetz, Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen: Beiträge zur GLDV-Tagung, Bonn, Germany (2005) 422–433.

[30] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching Word Vectors with Subword Information, Trans. Assoc. Comput. Linguistics 5 (2017) 135–146. doi:10.1162/tacl_a_00051.

[31] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised Cross-lingual Representation Learning at Scale, in: ACL 2020, Association for Computational Linguistics, 2020, pp. 8440–8451. doi:10.18653/v1/2020.acl-main.747.

[32] M. Konkol, M. Konopík, Segment Representations in Named Entity Recognition, in: Text, Speech, and Dialogue, Springer International Publishing, 2015, pp. 61–70. doi:10.1007/978-3-319-24033-6_7.

[33] T. Proisl, P. Uhrig, SoMaJo: State-of-the-art tokenization for German web and social media texts, in: WAC@ACL 2016, Association for Computational Linguistics, 2016, pp. 57–62. doi:10.18653/v1/W16-2607.

[34] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, R. Vollgraf, FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP, in: NAACL-HLT 2019, Association for Computational Linguistics, 2019, pp. 54–59. doi:10.18653/v1/n19-4010.

[35] A. Akbik, D. Blythe, R. Vollgraf, Contextual String Embeddings for Sequence Labeling, in: COLLING 2018, Association for Computational Linguistics, 2018, pp. 1638–1649. URL: https://aclanthology.org/C18-1139.

[36] S. Schweter, A. Akbik, FLERT: Document-Level Features for Named Entity Recognition (2020). doi:10.48550/ARXIV.2011.06993.

[37] B. Chan, S. Schweter, T. Möller, German's Next Language Model, in: COLING 2020, International Committee on Computational Linguistics, 2020, pp. 6788–6796. doi:10.18653/v1/2020.coling-main.598.