Investigation on Data Adaptation Techniques
for Neural Named Entity Recognition

Evgeniia Tokarchuk*, David Thulke†, Weiyue Wang†, Christian Dugast†, and Hermann Ney†

*Informatics Institute, University of Amsterdam
†Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
`e.tokarchuk@uva.nl`
`{thulke,wwang,dugast,ney}@cs.rwth-aachen.de`

Abstract

Data processing is an important step in various natural language processing tasks. As the commonly used datasets in named entity recognition contain only a limited number of samples, it is important to obtain additional labeled data in an efficient and reliable manner. A common practice is to utilize large monolingual unlabeled corpora. Another popular technique is to create synthetic data from the original labeled data (data augmentation). In this work, we investigate the impact of these two methods on the performance of three different named entity recognition tasks.

1 Introduction

Recently, deep neural network models have emerged in various fields of natural language processing (NLP) and replaced the mainstream position of conventional count-based methods (Lample et al., 2016; Vaswani et al., 2017; Serban et al., 2016). In addition to providing significant performance improvements, neural models often require high hardware conditions and a large amount of clean training data. However, there is usually only a limited amount of cleanly labeled data available, so techniques such as data augmentation and self-training are commonly used to generate additional synthetic data.

Significant progress has been made in recent years in designing data augmentations for computer vision (CV) (Krizhevsky et al., 2012), automatic speech recognition (ASR) (Park et al., 2019), natural language understanding (NLU) (Hou et al., 2018) and machine translation (MT) (Wang et al., 2018) in supervised settings. In addition, semi-supervised approaches using self-training techniques (Blum and Mitchell, 1998) have shown

promising performance in conventional named entity recognition (NER) systems (Kozareva et al., 2005; Daumé III, 2008; Täckström, 2012). In this work, the effectiveness of self-training and data augmentation techniques on neural NER architectures is explored.

To cover different data situations, we select three different datasets: The English CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) dataset, which is the benchmark on which almost all NER systems report results, it is very clean and the baseline models achieve an F1 score of around 92.6%; The English W-NUT 2017 (Derczynski et al., 2017) dataset, which is generated by users and contains inconsistencies, baseline models get an F1 score of around 52.7%; The GermEval 2014 (Benikova et al., 2014) dataset, a fairly clean German dataset with baseline scores of around 86.3%[1]. We observe that the baseline scores on clean datasets such as CoNLL and GermEval can hardly be improved by data adaptation techniques, while the performance on the W-NUT dataset, which is relatively small and inconsistent, can be significantly improved.

2 Related Work

2.1 State-of-the-art Techniques in NER

Collobert et al. (2011) advance the use of neural networks (NN) for NER, who propose an architecture based on temporal convolutional neural networks (CNN) over the sequence of words. Since then, many articles have suggested improvements to this architecture. Huang et al. (2015) propose replacing the CNN encoder in Collobert et al. (2011) with a bidirectional long short-term memory (LSTM) encoder, while Lample et al. (2016) and Chiu and Nichols (2016) introduce a hierarchy into the architecture by replacing artificially designed features

---

*Work completed while studying at RWTH Aachen University.

[1]From here on, for the sake of simplicity, we omit the annual information of the datasets.

with additional bidirectional LSTM or CNN encoders. In other related work, Mesnil et al. (2013) have pioneered the use of recurrent neural networks (RNN) to decode tags.

Recently, various pre-trained word embedding techniques have offered further improvements over the strong baseline achieved by the neural architectures. Akbik et al. (2018) suggest using pre-trained character-level language models from which to extract hidden states at the start and end character positions of each word to embed any string in a sentence-level context. In addition, the embedding generated by unsupervised representation learning (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019; Taillé et al., 2020) has been used successfully for NER, as well as other NLP tasks. In this work, the strongest model for each task is used as the baseline model.

2.2 Data Adaptation in NLP

In NLP, generating synthetic data using forward or backward inference is a commonly used approach to increase the amount of training data. In strong MT systems, synthetic data that is generated by back-translation is often used as additional training data to improve translation quality (Sennrich et al., 2016). A similar approach using backward inference is also successfully used for end-to-end ASR (Hayashi et al., 2018). In addition, back-translation, as observed by Yu et al. (2018), can create various paraphrases while maintaining the semantics of the original sentences, resulting in significant performance improvements in question answering.

In this work, synthetic annotations, which are generated by forward inference of a model that is trained on annotated data, are added to the training data. The method of generating synthetic data by forward inference is also called self-training in semi-supervised approaches. Kozareva et al. (2005) use self-training and co-training to recognize and classify named entities in the news domain. Täckström (2012) uses self-training to adapt a multi-source direct transfer named entity recognizer to different target languages, "relexicalizing" the model with word cluster features. Clark et al. (2018) propose cross-view training, a semi-supervised learning algorithm that improves the representation of a bidirectional LSTM sentence encoder using a mixture of labeled and unlabeled data.

In addition to the promising pre-trained embedding that is successfully used for various NLP tasks, the masked language modeling (MLM) can also be used for data augmentation. Kobayashi (2018) and Wu et al. (2019) propose to replace words with other words that are predicted using the language model at the corresponding position, which shows promising performance on text classification tasks. Recently, Kumar et al. (2020) discussed the effectiveness of such different pre-trained transformer-based models for data augmentation on text classification tasks. And for neural MT, Gao et al. (2019) suggest replacing randomly selected words in a sentence with a mixture of several related words based on a distribution representation. In this work, we explore the use of MLM-based contextual augmentation approaches for various NER tasks.

3 Self-training

Though, the amount of annotated training data is limited for many NLP tasks, additional unlabeled data is available in most situations. Semi-supervised learning approaches make use of this additional data. A common way to do this is self-training (Kozareva et al., 2005; Täckström, 2012; Clark et al., 2018).

At a high level, it consists of the following steps:

1. An initial model is trained using the labeled data.

2. This model is used to annotate the additional unlabeled data.

3. A subset of this data is selected and used in addition to the labeled data to retrain the model.

For the performance of the method it is critical to find a heuristic to select a good subset of the automatically labeled data. The selected data should not introduce too many errors, but at the same time they should be informative, i.e. they should be useful to improve the decision boundary of the final model. One selection strategy (Drugman et al., 2016) is to calculate a confidence measure for all unlabeled sentences and to randomly sample sentences above a certain threshold.

We consider two different confidence measures in this work. The first, hereinafter referred to as c_1, is the posterior probability of the tag sequence y given the word sequence x:

$$c_1(y, x) = p(y \mid x) = \frac{e^{s(x,y)}}{\sum_{y'} e^{s(x,y')}} \qquad (1)$$

whereby $s(x, y)$ is the unnormalized log score assigned by the model to the sequence, consisting of an emission model q_i^E and transition model q^T:

$$s(x, y_1^T) = \sum_{i=1}^{T} q_i^E(y_i \mid x) + q^T(y_i \mid y_{i-1})$$

For the second confidence measure, we take into account the normalized tag scores at each position. To get a confidence score for the entire sequence, we take the minimum tag score of all positions. Thus, c_2 is defined as follows:

$$c_2(y, x) = \min_i \frac{q_i^E(y_i \mid x) + q^T(y_i \mid y_{i-1})}{\sum_{y_i'} q_i^E(y_i' \mid x) + q^T(y_i' \mid y_{i-1})} \tag{2}$$

4 MLM-based Data Augmentation

Instead of using additional unlabeled data, we apply MLM-based data augmentation specifically for NER by masking and replacing original text tokens while maintaining labels.

For each masked token x_i:

$$\hat{x}_i = \arg\max_w p(x_i = w | \tilde{\mathbf{x}}) \tag{3}$$

where $\hat{x}_i$ is the predicted token, $w \in V$ is the token from the model vocabulary and $\tilde{\mathbf{x}}$ is the original sentence with $x_i = $ [MASK].

There are several configurations that can affect the performance of the data augmentation method: Techniques of selecting the tokens to be replaced, the order of token replacement in case of multiple replacement and the criterion for selecting the best tokens from the predicted ones. This section studies the effect of these configurations.

4.1 Sampling

Entity spans (entities of arbitrary length) make the training sentences used in NER tasks special. Since there is no guarantee that a predicted token belongs to the same entity type as an original token, it is important to ensure that the masked token is not in the middle of the entity span and that the existing label is not damaged. In this work, we propose three different types of token selection inside and outside of entity spans:

- **Entity replacement**: Collect entity spans of length one in the sentence and randomly select the entity span to be replaced. In this case, exactly one entity in the sentence is replaced. The sentences without entities or with longer entity spans are skipped.

- **Context replacement**: We consider tokens with the label "O" as context and alternate between two setups: (1) Select only context tokens before and after entities, and (2) select a random subset of context tokens among all context tokens.

- **Mixed**: Select uniformly at random the number of masked tokens between two and the sentence length among all tokens in the sentence.

The first approach allows only one entity to be generated and thus benefits from conditioning to the full sequence context. However, it does not guarantee the correct labeling for the generated token. The disadvantage of the second approach is that we do not generate new entity information, but only generate a new context for the existing entity spans. Even if a new entity type is generated, it has the original "O" label without a NER classification pipeline. The disadvantage of the third approach is that the token may be selected in the middle of the entity span and the label is no longer relevant. The sampling approaches depicted on the Figure 1. In addition, the number of replaced tokens should be properly tuned to avoid inadequate generation. In this work, we do not set any boundaries for maximum token replacement and leave such investigation to future work.

4.2 Order of Generation

In our method, we predict exactly one mask token per time. Our sampling approaches allow multiple tokens to be replaced. Therefore we have two possible options for the generation order:

- **Independent**: Each consecutive masking and prediction is made on top of the original sequence.

- **Conditional**: Each consecutive masking and prediction is made on top of the prediction of the previous step.

4.3 Criterion

The criterion is an important part of the generation process. On the one hand, we want our synthetic sequence to be reliable (highest token probability), on the other hand, it should differ as much as possible from the original sequence (high distance). We

---

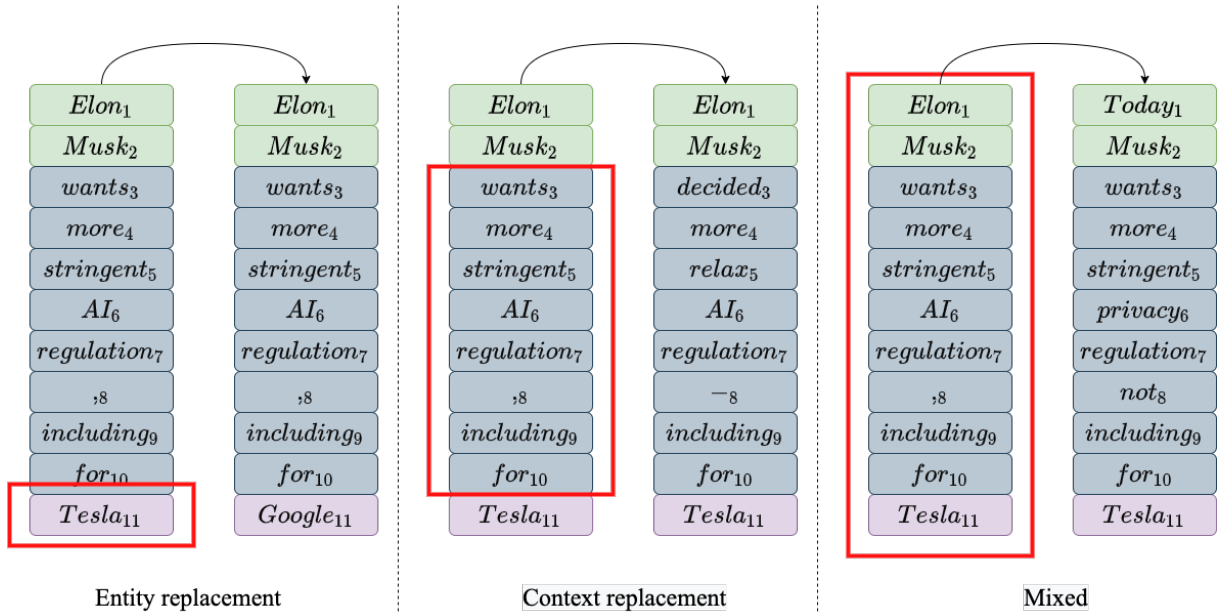[2]Given example is taken from https://artificialintelligence-news.com

Figure 1: Sampling approaches example[2] for the MLM data augmentation. Gray color refers to the tokens with the entity type "O" (context), green color refers to the `PER` entity type and purple color refers to the `ORG` entity type. Red square represents the subset of tokens which is used for replacement.

propose two criteria for choosing the best token from the five-best predictions:

- **Highest probability (top token)**: Choose the target token only based on the MLM probability for that token.

- **Highest probability and distance (joint criterion)**: Choose the target token based on the product of the MLM probability for the token and Levenshtein distance (Levenshtein, 1966) between the original sentence and the sentence with the new token.

Regardless of the combination of the parameters, the sentences must be changed. As a result, we guarantee that there is no duplication in our synthetic data with the original dataset.

4.4 Discussion

The main disadvantage of using a language model (LM) for the augmentation of NER datasets is that the LM does not take into account the labeling of the sequence and the prediction of the masked token, which only depends on the surrounding tokens. As a result, we lose important information for decision-making. Incorporating label information as described in Wu et al. (2019) into the MLM would be the way to tackle this problem.

Another way to reduce the noise in the generated dataset is to apply a filtering step to the generation

pipeline. One way to incorporate filtering into the augmentation process is to set the threshold for the MLM token probabilities: If the probability of the predicted token is less than a threshold, we ignore such prediction. However, the problem of misaligning token labels is not resolved. Therefore, we adapt our proposed confidence measure from Section 3 for filtering.

In this work, we do not discuss the selection of the MLM itself as well as the effects of fine-tuning on the specific task.

5 Experiments

5.1 Datasets

We test our data adaptation approaches with three different NER datasets: CoNLL (Tjong Kim Sang and De Meulder, 2003), W-NUT (Derczynski et al., 2017) and GermEval (Benikova et al., 2014).

All datasets have the original labeling scheme as `BIO`, but following Lample et al. (2016) we convert it to the `IOBES` scheme for training and evaluation. For our baseline models, we do not use any additional data apart from the provided training data. Development data is only used for validation. For CoNLL we skip all document boundaries. The statistics for the datasets are shown in Table 1.[3]

---

[3]Further details on the used datasets can be found in Appendix A

4

Dataset	train	dev	test
CoNLL	14041	3250	3453
W-NUT	3394	1008	1287
GermEval	24001	2199	5099

Table 1: Dataset sizes in number of sentences.

5.2 Model Description

The Bidirectional LSTM - Conditional Random Field (BiLSTM-CRF) model (Lample et al., 2016) is a widely used architecture for NER tasks. Together with pre-trained word embeddings, it surpasses other neural architectures. We use the BiLSTM-CRF model implemented in the *Flair*[4] framework version 0.5, which delivers the state-of-the-art performance.

The BiLSTM-CRF model consists of 1 hidden layer with 256 hidden states. Following Reimers and Gurevych (2017), we set the initial learning rate to 0.1 and the mini-batch size to 32. For each task, we select the best performing embedding from all embedding types in *Flair*. For training models with CoNLL data, we use pre-trained *GloVE* (Pennington et al., 2014) word embedding (Grave et al., 2018) together with the Flair embedding (Akbik et al., 2018) as input into the model. For W-NUT experiments, we use *roberta-large* embedding provided by *Transformers* library (Wolf et al., 2019). German *dbmdz/bert-base-german-cased* embedding is used for experiments with the GermEval dataset.

5.3 Unlabeled Data

Additional unlabeled data is required for self-training. To match the domain of the test data, we collect the data from the sources mentioned in the individual task descriptions.

W-NUT Like the test data, the data for W-NUT consists of user comments from Reddit, which were created in April 2017[5] (comments in the test data were created from January to March 2017), as well as titles, posts and comments from StackExchange, which were created from July to December 2017[6] (the content of the test data was created from January to May 2017). The documents are filtered

according to length and community as described in the task description paper and tokenized with the *TweetTokenizer* from *nltk*[7].

CoNLL The data was sampled from news articles in the Reuters corpus from October and November 1996. The sentences are tokenized using *spaCy*[8] and filtered (by removing common patterns like the date of the article, sentences that do not contain words and sentences with more than 512 characters as this is the length of the longest sentence in the CoNLL training data).

GermEval We randomly sampled additional data from sentences extracted from news and Wikipedia articles provided by the Leipzig Corpora Collection[9]. In addition to tokenizing the sentences using spaCy, we do not do any additional preprocessing or filtering.

5.4 Self-training

Before applying the approach described in Section 3, we need to find the thresholds t for the confidence measures c_1 and c_2 for each corpus. We evaluate both confidence measures on the development sets of the three corpora. One way to evaluate confidence measures is to calculate the confidence error rate (CER). It is defined as the number of misassigned labels (i.e. confidence is above the threshold and the prediction of the model is incorrect or the confidence is below the threshold and the prediction is correct) divided by the total number of samples.

Figure 2 shows the CER of c_1 and c_2 on the development set of W-NUT for different threshold values t. For the threshold of 0.0 or 1.0 the CER degrades to the percentage of incorrect or correct predictions as either all or no confidence values are above the threshold. For c_2 there is a clear optimum at $\hat{t}_2 = 0.42$ and for larger and smaller thresholds the CER rises rapidly.

In contrast, the optimum for c_1 at $\hat{t}_1 = 0.57$ is not as pronounced. This motivated us not only to choose the best value in terms of CER, but also a lower threshold $t'_1 = 0.42$ with slightly worse CER. In this way, we include more sentences where the model is less confident without introducing too many additional errors. The threshold values for

---

[4]https://github.com/zalandoresearch/flair/
[5]https://files.pushshift.io/reddit/comments/
[6]https://archive.org/download/stackexchange

[7]https://www.nltk.org/api/nltk.tokenize.html
[8]https://github.com/explosion/spaCy
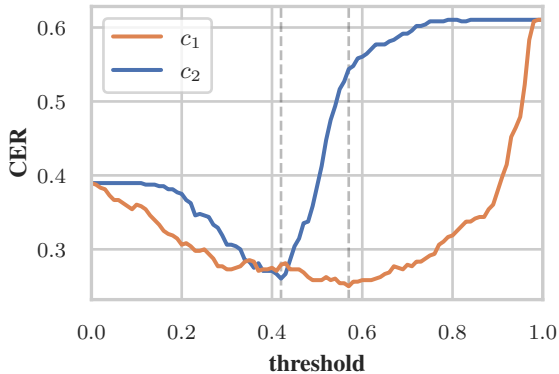[9]https://wortschatz.uni-leipzig.de/de/download

Figure 2: CERs for c_1 (orange) and c_2 (blue) with different threshold values on the W-NUT development set. Vertical dashed lines represent $\hat{t}_1$ and $\hat{t}_2$.

	W-NUT	CoNLL	GermEval
$\hat{t}_1$	0.57	0.83	0.63
t'_1	0.42	0.70	0.50
$\hat{t}_2$	0.42	0.50	0.47

Table 2: Selected confidence threshold values.

CoNLL and GermEval are selected analogously. Table 2 provides an overview of all threshold values that are used in all subsequent experiments.

The unlabeled data is annotated using the baseline models described in Section 3 (we choose the best runs based on the score on the development set) and is filtered based on the different confidence thresholds. Then we sample a random subset of size k from these remaining sentences. For tasks where the data comes from different sources, e.g. news and Wikipedia for GermEval, we uniformly sample from the different sources to avoid that a particular domain is overrepresented. The selected additional sentences are then appended to the original set of training sentences to create a new training set that is used to retrain the model from scratch.

To validate our selection strategy, we test our pipeline with different confidence thresholds for both confidence measures. Figure 3 shows the results on the test set of W-NUT. For each threshold, 3394 sentences are sampled, i.e. the size of the training set is doubled. The results confirm our selection strategy. t'_1 and $\hat{t}_2$ give the best results of all tested threshold values. In particular, t'_1 performs better than $\hat{t}_1$.

Table 3 shows the results of self-training on all three datasets. For each of them, we test the three selection strategies by sampling new sentences in the size of 0.5 times, 1 times and 2 times the size of
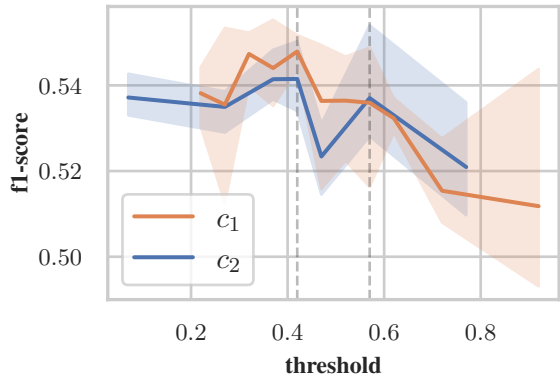

Figure 3: Average F1 scores and standard deviation (shaded area) of 3 runs on the test set of W-NUT after retraining the model on additional data selected using different confidence measures (color) and thresholds.

the original training data. For W-NUT we get up to 2% of the absolute improvements in the F1 score over the baseline. On larger datasets like CoNLL and GermEval these effects disappear and we only get improvements of up to 0.1% and in some cases even deterioration.

5.5 MLM-based Data Augmentation

We follow the approach explained in Section 4 and generate synthetic data using pre-trained models from the Transformers library. We concatenate original and synthetic data and train the NER model on the new dataset. We test all possible combinations of the augmentation parameters from Section 4 on the W-NUT dataset. Table 4 shows the result of the augmentation. When sampling with one entity, there is no difference between independent and conditional generation, since only one token in a sentence is masked. We therefore only carry out an independent generation for this type of sampling. We report an average result among 3 runs along with a standard deviation of the model with different random seeds.

W-NUT and CoNLL datasets are augmented using a pre-trained English BERT model[10] and GermEval with a pre-trained German BERT model[11] respectively. We do not fine-tune these models.

Sampling from the context of the entity spans shows significant improvements on W-NUT test set. First of all, it includes implicit filtering: Only the sentences with the entities are selected and re-

---

[10]https://huggingface.co/
bert-large-cased-whole-word-masking
[11]https://huggingface.co/
bert-base-german-cased

		W-NUT		CoNLL		GermEval	
		Δ sen.	F1	Δ sen.	F1	Δ sen.	F1
1	baseline	+0%	52.7 ± 2.48	+0%	92.6 ± 0.18	+0%	86.3 ± 0.06
2	$c_1 \geq \hat{t}_1$	+50%	54.2 ± 0.35	+50%	92.5 ± 0.06	+50%	86.0 ± 0.08
3	$c_1 \geq \hat{t}_1$	+100%	53.6 ± 1.41	+100%	92.5 ± 0.12	+100%	86.1 ± 0.26
4	$c_1 \geq \hat{t}_1$	+200%	53.5 ± 0.53	+200%	92.4 ± 0.08	+200%	86.3 ± 0.14
5	$c_1 \geq t'_1$	+50%	53.7 ± 1.95	+50%	92.5 ± 0.02	+50%	86.1 ± 0.21
6	$c_1 \geq t'_1$	+100%	$\mathbf{54.8 \pm 0.33}$	+100%	92.6 ± 0.09	+100%	86.2 ± 0.12
7	$c_1 \geq t'_1$	+200%	53.5 ± 0.29	+200%	92.5 ± 0.06	+200%	$\mathbf{86.4 \pm 0.03}$
8	$c_2 \geq \hat{t}_2$	+50%	54.6 ± 0.42	+50%	$\mathbf{92.7 \pm 0.04}$	+50%	86.0 ± 0.16
9	$c_2 \geq \hat{t}_2$	+100%	54.2 ± 0.98	+100%	92.6 ± 0.06	+100%	$\mathbf{86.4 \pm 0.15}$
10	$c_2 \geq \hat{t}_2$	+200%	54.5 ± 0.43	+200%	$\mathbf{92.7 \pm 0.02}$	+200%	86.3 ± 0.05

Table 3: Results of self-training.

placed. Therefore, compared to other methods, we add less new sentences (except when replacing entities). Second of all, since replacing tokens with a language model should result in the substitution with similar words, the label is less likely to be destroyed while context tokens are replaced.

On the other hand, the mixed sampling strategy performs the worst among all methods. We believe that this is the effect when additional noise is included in the dataset (by noise we mean all types of noise, e.g. incorrect labeling, grammatical errors, etc). Allowing masking of words up to sequence in some cases destroys the sentence, e.g. incorrect and multiple occurrences of the same words can occur. In Appendix B we present the examples of augmented sentences for each augmentation approach and each dataset. Additionally, we report the average number of masked token.

To analyze the resulting models, we plot the average confidence scores of the test set as well as the number of errors per sentence for the best baseline model and best augmented model. We use the best baseline system with 54.6% F1 score and the best model corresponding to the setup of line 8 in Table 4 with 57.4% F1 score. We count the error every time the model predicts a correct label with low confidence or an incorrect label with high confidence. We set high and low confidence to be 0.6 and 0.4 respectively. Figure 4 shows that the augmented model makes a more reliable prediction than the best baseline system model.

We repeat the promising MLM generation pipeline on the CoNLL and GermEval datasets. These datasets contain more entities in the original data. In addition, even though the entity replacement sampling did not work well on W-NUT
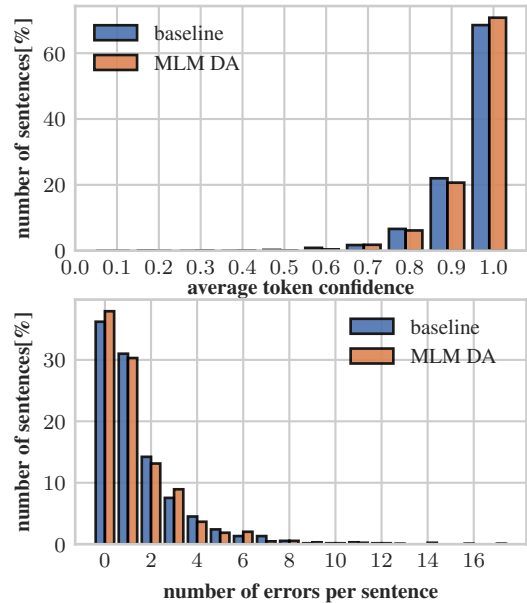

Figure 4: Average confidence score and the error per sentence on W-NUT test data. `MLM DA` refers to the setup of line 8 in Table 4

dataset, we repeat these experiments, since generating new entities is the most interesting scenario for using the MLM augmentation.

Although the MLM-based data augmentation leads to improvements of up to 3.6% F1 score on the W-NUT dataset, Table 5 shows that such effect disappears when we apply our method to larger and cleaner datasets such as CoNLL and GermEval. We believe there are several reasons for that. First, our MLM-based data augmentation method does not guarantee the accuracy of the labeling after augmentation. So for larger datasets, there are many more possibilities to increase the noise of the corpus. Moreover, we do not study

		sampling	generation	criterion	Δ sen.	F1
1	baseline	-	-	-	+0.0%	52.7 ± 2.48
2	MLM DA	entity	independent	top token	+24.4%	53.7 ± 0.91
3				joint	+24.7%	54.6 ± 0.50
4		mixed	conditional	top token	+98.7%	52.3 ± 1.25
5				joint	+99.7%	51.7 ± 1.36
6			independent	top token	+98.6%	53.7 ± 0.89
7				joint	+99.7%	53.3 ± 0.61
8		context	conditional	top token	+33.8%	**56.3 ± 1.21**
9				joint	+35.8%	**55.6 ± 1.12**
10			independent	top token	+33.8%	**55.0 ± 1.16**
11				joint	+35.8%	**56.0 ± 0.06**
12		random context	conditional	top token	+96.8%	**54.9 ± 0.40**
13				joint	+99.7%	54.5 ± 1.21
14			independent	top token	+96.9%	53.7 ± 0.93
15				joint	+99.7%	53.5 ± 2.40

Table 4: Results of the MLM-based augmentation on the W-NUT dataset. `entity` refers to the sampling tokens from entity spans of length one, `mixed` means sampling from the complete sequence, `context` indicates sampling from the entity span context, `random context` denotes sampling from random context labels. `conditional` refers to the conditional generation and `independent` refers to the independent generation type. The `top token` criterion selects the token based on the highest probability, and the `joint` criterion takes into account the token probability and the Levenshtein distance.

how well pre-trained models suit the specific task, which might be crucial for the DA. Besides, for GermEval augmentation, we use the BERT model with three times fewer parameters than for W-NUT and CoNLL.

5.5.1 Filtering of Augmented Data

As discussed in Section 4, an additional data filtering step can be applied on top of the augmentation process. We report results on two different filtering methods: First, we set a threshold for the probability of the predicted token (in our experiments we use the probability 0.5); Second, we filter sentences by minimum confidence scores as discussed in Section 3. We set the minimum confidence score according to Table 2. We apply filtering to the worst and best-performing model according to the numbers in Table 4. The filtering results on W-NUT are shown in Table 6.

In the case of the worst model, filtering based on the token probability improve the performance of the model by 2.6% compared to the unfiltered one. Filtering by confidence score does not improve the performance, but significantly reduces the standard deviation of the score. The results are expected, since by using token probability we increase the sentence reliability and completely change the synthetic data, while using the confidence score we

filter on the same synthetic data. In the case of the better model, we see the opposite trend. Here filtering leads to performance degradation and an increase in the standard deviation.

We apply the same filtering techniques for CoNLL and GermEval. Table 7 shows the results for 3 different models. We choose the best, the worst and the model with the highest number of additional sentences for filtering. In the case of the worst model, the performance is improved by 1.1% F1 score with the minimum confidence filtering for CoNLL and 0.5% F1 score for GermEval compared to the unfiltered version. However, for the best model, the results remain at the same level and the baseline systems are not improved.

Although we do not achieve significant improvements compared to the baseline system, we see a potential in the MLM-based augmentation with the combination with filtering.

6 Discussion and Future Work

In this work, we present results of data adaptation methods on various NER tasks. We show that MLM-based data augmentation and self-training approaches lead to improvements on the small and noisy W-NUT dataset.

We propose two different confidence measures for self-training and empirically estimate the best

		sampling	generation	criterion	CoNLL		GermEval	
					Δ sen.	F1	Δ sen.	F1
1	baseline	-	-	-	+0.0%	**92.6 ± 0.18**	0.0%	**86.3 ± 0.06**
3	MLM DA	entity	independent	joint	+57.9%	91.5 ± 0.10	+47.9%	85.9 ± 0.06
8		context	conditional	top token	+65.7%	92.4 ± 0.12	+51.4%	86.1 ± 0.26
9				joint	+72.2%	92.3 ± 0.06	+58.5%	86.0 ± 0.15
10			independent	top token	+65.7%	92.5 ± 0.06	+51.4%	86.1 ± 0.15
11				joint	+72.2%	92.2 ± 0.17	+58.5%	86.0 ± 0.20
12		rand. cont.	conditional	top token	+85.1%	92.1 ± 0.15	+94.1%	86.1 ± 0.10

Table 5: Results of the MLM-based data augmentation on CoNLL and GermEval datasets. The row numbers refer to the row numbers of the Table 4.

	Δ sen.	filtering	F1
5	+99.7%	-	51.7 ± 1.36
	+86.3%	token prob.	**54.3 ± 0.31**
	+59.5%	min. conf.	51.2 ± 0.60
9	+33.8%	-	**56.3 ± 1.21**
	+13.8%	token prob.	53.3 ± 2.00
	+10.4%	min. conf.	51.7 ± 2.10

Table 6: F1 scores of using filtered augmented data on W-NUT. The row numbers refer to the row numbers of the Table 4.

		CoNLL		GermEval	
	filtering	Δ sen.	F1	Δ sen.	F1
3	none	+57.9%	91.5 ± 0.10	+47.9%	85.9 ± 0.06
	tok. prob.	+7.8%	92.4 ± 0.15	+13.1%	86.1 ± 0.29
	min. conf.	+13.5%	92.6 ± 0.15	+13.9%	**86.4 ± 0.12**
10	none	+65.7%	92.5 ± 0.06	+51.5%	86.1 ± 0.15
	tok. prob.	+22.5%	92.5 ± 0.15	+34.5%	**86.3 ± 0.21**
	min. conf.	+52.1%	**92.6 ± 0.20**	+23.9%	86.1 ± 0.10
12	none	+85.1%	92.1 ± 0.15	+94.1%	86.1 ± 0.10
	tok. prob.	+42.5%	**92.8 ± 0.06**	+76.1%	86.1 ± 0.00
	min. conf.	+58.9%	**92.6 ± 0.12**	+62.3%	86.0 ± 0.21

Table 7: F1 scores of using filtered augmented data on CoNLL and GermEval. The first line represents the augmentation method from Table 4.

thresholds. Our results on the W-NUT dataset show the effectiveness of the selection strategies based on those confidence measures.

For MLM-based data augmentation, we suggest multiple ways of generating synthetic NER data. Our results show that even without generating new entity spans we are able to achieve better results.

For future work, we would like to incorporate label information into the augmentation pipeline by either conditioning the token predictions on labels or adding additional classification steps on top of the token prediction. Another important question is the choice of the MLM and the impact of task-specific fine-tuning. Further investigations into the filtering step should also be carried out.

For both self-training and MLM-based data aug-

mentation we would like to improve the integration in the training process. The contribution of the original training data to the loss function could be increased or additional data could be weighted by their confidence. Finally, we would like to test whether we can combine the two methods to achieve additional improvements.

Acknowledgements

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 1638–1649, Santa Fe, NM, USA.

Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Padó. 2014. Germeval 2014 named entity recognition: Companion paper. *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition, Hildesheim, Germany*, pages 104–112.

Avrim Blum and Tom M. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998*, pages 92–100. ACM.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Hal Daumé III. 2008. Cross-task knowledge-constrained self training. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 680–688, Honolulu, Hawaii. Association for Computational Linguistics.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

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

Thomas Drugman, Janne Pylkkönen, and Reinhard Kneser. 2016. Active and semi-supervised learning in asr: Benefits on the acoustic and language models. In *Interspeech 2016*, pages 2318–2322.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544, Florence, Italy. Association for Computational Linguistics.

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

Tomoki Hayashi, Shinji Watanabe, Yu Zhang, Tomoki Toda, Takaaki Hori, Ramón Fernández Astudillo, and Kazuya Takeda. 2018. Back-translation-style data augmentation for end-to-end ASR. In *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, pages 426–433. IEEE.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1234–1245. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Zornitsa Kozareva, Boyan Bonev, and Andres Montoyo. 2005. Self-training and co-training applied to spanish named entity recognition. In *Proceedings of the 4th Mexican International Conference on Artificial Intelligence*, pages 770–779, Monterrey, Mexico.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris" Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 260–270, San Diego, CA, USA.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH 2013, 14th Annual Conference of the International*

Speech Communication Association, Lyon, France, August 25-29, 2013, pages 3771–3775. ISCA.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 2613–2617. ISCA.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

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

Oscar Täckström. 2012. Nudging the envelope of direct transfer methods for multilingual named entity recognition. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 55–63, Montréal, Canada. Association for Computational Linguistics.

Bruno Taillé, Vincent Guigue, and Patrick Gallinari. 2020. Contextualized embeddings in named-entity recognition: An empirical study on generalization. In *Advances in Information Retrieval*, pages 383–391, Cham. Springer International Publishing.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 142–147, Edmonton, Canada.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 5998–6008, Long Beach, CA, USA.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional BERT contextual augmentation. In *Computational Science - ICCS 2019 - 19th International Conference, Faro, Portugal, June 12-14, 2019, Proceedings, Part IV*, volume 11539 of *Lecture Notes in Computer Science*, pages 84–95. Springer.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

A   Data Description

In our work we use three NER datasets:

- CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) contains news articles from the Reuters[12] corpus. The annotation contains 4 entity types `person`, `location`, `organization`, `miscellaneous`. We remove the document boundary information for our experiments.

- W-NUT 2017 (Derczynski et al., 2017) contains texts from Twitter (training data), YouTube (development data), StackExchange and Reddit (test data). The annotation contains 6 entity types: `person`, `location`, `corporation`, `product`, `creative-work`, `group`

- GermEval 2014 (Benikova et al., 2014): contains the data from the German Wikipedia and news Corpora. The annotation contains 12 entity types: `location`, `organization`, `person`, `other`, `location deriv`, `location part`, `organization deriv`, `organization part`, `person deriv`, `person part`, `other deriv`, `other part`.

Table 8 shows detailed statistics of those datasets. Together with number of entities, tokens and sentences we report the percentage of the labelled tokens among all the tokens.

Dataset		train	dev	test
CoNLL	#sentences	14041	3250	3453
	#entities	23500	5943	5649
	#tokens	203621	51362	46435
	#entity types	4	4	4
	%labelled	16.7	16.8	17.5
W-NUT	#sentences	3394	1008	1287
	#entities	1976	836	1080
	#tokens	62730	15723	23394
	#entity types	6	6	6
	%labelled	5.0	7.9	7.4
GermEval	#sentences	24001	2199	5099
	#entities	29077	2674	6178
	#tokens	452790	41635	96475
	#entity types	12	12	12
	%labelled	9.3	9.5	9.3

Table 8: Dataset sizes in number of sentences, tokens and entities. Here, entity means the entity span, e.g. `European Union` is considered as one entity.

---

[12] https://trec.nist.gov/data/reuters/reuters.html

B   MLM-based Data Augmentation

B.1   Data statistics

The number of masked tokens solely depends on the augmentation strategy discussed in section 4. Table 9 reports the average number of masked tokens in the sentence on W-NUT dataset for each augmentation strategy. Table 10 and Table 11 show the average number of masked tokens in the sentence for the most promising augmentation strategies for CoNLL and GermEval tasks.

sampling	generation	criterion	Δ sen.	Masked
entity	independent	top token	+24.4%	1.2
		joint	+24.7%	1.2
mixed	conditional	top token	+98.7%	7.4
		joint	+99.7%	8.8
	independent	top token	+98.6%	7.0
		joint	+99.7%	8.8
context	conditional	top token	+33.8%	4.4
		joint	+35.8%	4.5
	independent	top token	+33.8%	4.3
		joint	+35.8%	4.5
random context	conditional	top token	+96.8%	7.1
		joint	+99.7%	8.1
	independent	top token	+96.9%	6.9
		joint	+99.7%	8.1

Table 9: Average number of masked tokens for each augmentation strategy on W-NUT dataset.

sampling	generation	criterion	Δ sen.	Masked
entity	independent	joint	+57.9%	1.1
context	conditional	top token	+65.7%	3.4
		joint	+72.2%	6.4
	independent	top token	+65.7%	3.4
		joint	+72.2%	6.4
random context	conditional	top token	+85.1%	4.5

Table 10: Average number of masked tokens on CoNLL dataset.

sampling	generation	criterion	Δ sen.	Masked
entity	independent	joint	+47.9%	1.0
context	conditional	top token	+51.4%	4.4
		joint	+58.5%	5.7
	independent	top token	+51.4%	4.3
		joint	+58.5%	5.3
random context	conditional	top token	+94.1%	6.0

Table 11: Average number of masked tokens on GermEval dataset.

B.2   Data Examples

We show the data examples on different dataset by varying one augmentation parameter while keeping others unchanged. Table 12 shows the examples on W-NUT dataset. In Table 13 and Table 14 we collect the examples for GermEval and CoNLL.

Parameter	Value	Example
Sampling	-	RT @Quotealicious: Today, I saw a guy driving a <corporation>Pepsi</corporation> truck, drinking a <product>Coke</product>. MLIA #Quotealicious
	entity	RT @Quotealicious: Today, I saw a guy driving a <corporation>Pepsi</corporation> truck, drinking a **<product>beer</product>** MLIA #Quotealicious
	context	RT @Quotealicious : Today, I saw a guy driving a <corporation>Pepsi</corporation> **car,** drinking a <product>Coke</product>. MLIA #Quotealicious
	random context	**m me:** Today, I saw a **man** driving a <corporation>Pepsi</corporation> truck, **buying** a <product>Coke</product>. MLIA #Quotealicious
	mixed	**m** @Quotealicious **Earlier** Today, I saw a guy driving a <corporation>Pepsi</corporation> truck, drinking a <product>Coke</product>. MLIA #Quotealicious
Order	-	What is everyone watching this weekend? <group>Twins</group>? <group>Vikings</group>? anyone going to see <creativework>Friday Night Lights</creativework>?
	independent	What is everyone watching this weekend? <group>Twins</group>? <group>Vikings</group>? anyone going to see <creativework>**the** Night Lights</creativework>?
	conditional	What is **he** doing this weekend with **<group>the</group> ##ing** <group>Vikings</group>? anyone going to install <creativework>Friday Night **lights**</creativework>?
Criterion	-	<person>Oscar</person>'s new favorite pass time is running as fast as he can from one end of the house to another yelling BuhBYYYYYE
	top token	**<person>Jack</person>**'s new favorite pass time is running as fast as he can from one end of the house to another yelling BuhBYYYYYE
	joint	**<person>Ben</person>**'s new favorite pass time is running as fast as he can from one end of the house to another yelling BuhBYYYYYE

Table 12: Data examples of W-NUT augmentation.

Parameter	Value	Example
Sampling	-	Zu einer Gebietsveränderung kam es 1822, als das vorher selbständige `<LOC>Champsigna</LOC>` nach `<LOC>Soucia</LOC>` eingemeindet wurde.
	entity	Zu einer Gebietsveränderung kam es 1822, als das vorher selbständige `<LOC>Champsigna</LOC>` nach **`<LOC>Paris</LOC>`** eingemeindet wurde.
	context	Zu einer Gebietsveränderung kam es 1822, als das vorher selbständige `<LOC>Champsigna</LOC>` nach `<LOC>Soucia</LOC>` **verlegt** wurde.
	random context	Zu einer Gebietsveränderung kam es 1822, als das **damals** selbständige `<LOC>Champsigna</LOC>` nach `<LOC>Soucia</LOC>` eingemeindet wurde.
	mixed	Zu einer **Eingemeindung** kam es 1822, als **die damals** selbständige **`<LOC>Dorf</LOC>`** nach **`<LOC>Turin</LOC>` verlegt** wurde.
Order	-	Aus diesem Grund wurde er Anfang Januar auch nach nur wenigen Tagen aus dem Klinikum `<LOC>Jena</LOC>` in eine Reha-Einrichtung am `<LOC>Bodensee</LOC>` verlegt.
	independent	**Zu** diesem Grund wurde er Anfang Januar **und** nach nur **zwei** Tagen aus dem Klinikum `<LOC>Jena</LOC>` in **die** Reha-Einrichtung am **`<LOC> Boden </LOC>`** verlegt.
	conditional	Aus diesem Grund **wo ich** Anfang Januar auch nach nur wenigen Tagen aus dem Klinikum `<LOC>Jena</LOC>` in **die** Reha-Einrichtung am `<LOC>Bodensee</LOC>` verlegt.
Criterion	-	Mit ihm der gleichen Meinung sind `<PER>Pyrrhon</PER>` und `<PER>Erillus</PER>` von `<LOC>Karthago</LOC>`.
	top token	Mit ihm der gleichen Meinung sind `<PER>Pyrrhon</PER>` und **`<PER>Gregor</PER>`** von `<LOC>Karthago</LOC>`.
	joint	Mit ihm der gleichen Meinung sind **`<PER>Alexander</PER>`** und `<PER>Erillus</PER>` von `<LOC>Karthago</LOC>`.

Table 13: Data examples of GermEval augmentation.

Parameter	Value	Example
Sampling	-	`<PER>Christopher Reeve</PER> -- <PER>Reeve</PER> was best known for playing the comic book hero <PER>Superman</PER> in four movies but his greatest heroics came in real life.`
	entity	`<PER>Christopher Reeve</PER> -- <PER>Reeve</PER> was best known for playing the comic book hero **<PER>Batman</PER>** in four movies but his greatest heroics came in real life .`
	context	`<PER>Christopher Reeve</PER> **The** <PER>Reeve</PER> **is** best known for playing the comic book superhero <PER>Superman</PER> in four movies but his greatest heroics came in real life.`
	random context	`<PER>Christopher Reeve</PER> -- <PER>Reeve</PER> **popular** best known for **popular popular popular** book hero <PER>Superman</PER> in four movies but his popular heroics came in real **popular popular**`
	mixed	`<PER>Christopher Reeve</PER> **The** **<PER>He</PER> is** best known for playing the comic book superhero <PER>Superman</PER> in **the films** but his greatest heroics came in real life.`
Order	-	`Four weeks ago <ORG>Stagecoach </ORG> said it had agreed the deal in principle, and it expected to pay 110 million stg-plus for the firm, with <ORG>Swebus</ORG>' current owner, the state railway company.`
	independent	`Four **days** ago <ORG>it</ORG> said it had **made** the deal in principle, and it expected to **raise** 110 million **euros to** the **operation contract including** <ORG>Swebus</ORG> ' current **employer being** the state railway company.`
	conditional	`**Two years** ago <ORG>Stagecoach</ORG> said it had **made** the deal in principle, and **was** expected to pay 110 million **marks** for the **operation**, with <ORG>Swebus</ORG>**'s** owner, the **Swedish** railway company.`
Criterion	-	`<ORG>ZDF</ORG> said <LOC> Germany </LOC> imported 47,600 sheep from <LOC> Britain </LOC> last year, nearly half of total imports.`
	top token	`**<ORG>He</ORG>** said **<LOC> they </LOC>** imported **more goods** from **<LOC> Germany </LOC> that** year, nearly half of **all number.**
	joint	`<ORG>ZDF</ORG> **this <LOC> this </LOC> this** 47,600 sheep **this <LOC> this </LOC> this** year **this** nearly half of **this** imports.`

Table 14: Data examples of CoNLL augmentation.