NTCIR 17 Conference: Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan
DOI: https://doi.org/10.20736/0002001295

HPIDHC at NTCIR-17 MedNLP-SC: Data Augmentation and Ensemble Learning for Multilingual Adverse Drug Event Detection

Smilla Fox* Martin Preiß*
Florian Borchert Aadil Rasheed Matthieu-P. Schapranow
Hasso Plattner Institute for Digital Engineering, University of Potsdam
Potsdam, Germany
{smilla.fox|martin.preiss}@student.hpi.de
{florian.borchert|aadil.rasheed|schapranow}@hpi.de

ABSTRACT

The Social Media Adverse Drug Event Detection (SM-ADE) track of the NTCIR-17 MedNLP-SC shared task aims to identify adverse drug events (ADE) in Japanese, English, French, and German social media texts. In this paper, we describe selected details of our contribution addressing the shared task. As a base model, we fine-tune RoBERTa models for the different language subtasks. In addition, we apply ensemble learning and data augmentation techniques. By leveraging data augmentation, we successfully elevate the resulting micro-averaged F_1 scores on the German dataset by 5 *pp* compared to the baseline. The application of ensemble learning yields a remarkable improvement of 7 *pp*. Through combining RoBERTa with these methods, we achieve promising results in the challenge. Our best runs accomplish exact accuracy scores between 0.84 and 0.87 and per-class F_1 scores between 0.77 and 0.82, consistently achieving the second-best results across all languages.

KEYWORDS

Natural Language Processing, Social Media, Adverse Drug Event Detection, Data Augmentation, Ensemble Learning, Multilingual

TEAM NAME

HPIDHC

SUBTASKS

SM-ADE-JA
SM-ADE-EN
SM-ADE-DE
SM-ADE-FR

1 INTRODUCTION

The NTCIR-17 Medical Natural Language Processing for Social Media and Clinical Texts (MedNLP-SC) shared task investigates different medical applications for natural language processing (NLP). In this work, we propose an approach for the Social Media Adverse Drug Event Detection subtask [14]. The goal of this task is to identify adverse drug events (ADEs) in short messages such as tweets from X (formerly known as Twitter). This is a promising direction to identifying sofar unknown side effects, e.g., for recently introduced drugs, as an increasing number of people use the internet to share details about their personal experiences.

We frame ADE detection problem as a multi-label classification task, where the input is a text snippet and the output consists of labels for 22 ADE classes. To this end, it is crucial that the model can identify the different symptoms, but also understand whether they are indeed ADEs caused by medication or, for instance, just symptoms which caused the medication usage in the first place. Our method builds upon a simple Transformer-based classification model, using a pre-trained RoBERTa encoder [9]. This is similar to the baseline proposed by the task organizers [14]. We extend this approach through additional data augmentation and ensemble learning steps. For augmentation, we generate additional tweets using GPT-3.5, an instruction-tuned large language model (LLM) [11]. Furthermore, we combine several models using ensemble learning. We evaluate these two techniques individually and in combination. The methods were applied for all languages relevant to the task, i.e., German, English, Japanese, and French. We submitted predictions obtained using these methods on the held-out test set of the tasks and discuss the official results in this paper.

The remainder of this work is structured as follows: In section 2, we set our work in the context of related work. In section 3, we present the datasets we use and we share details about our incorporated methodology in section 4. We describe our experimental setup in section 5 and present the acquired results in section 6. We discuss our results in section 7 and conclude our work in section 8.

2 RELATED WORK

A number of prior work exists, which concern ADE detection in medical text documents. Murphy et al. [10] review 29 articles dealing with text from electronic health records in survey on supervised NLP methods for this task. Furthermore, Lardon et al. [7] reviewed ADE detection in social media text already in 2015. More recently, Raithel et al. [12] have investigated detection of ADE from a German patient online forum.

*Smilla Fox and Martin Preiß share first authorship.

Numerous approaches for detecting ADEs using NLP focus on the simple binary classification task of determining whether a text contains ADE mentions or not. There are two main types of texts that are usually used as input.

Lately, the problem has been mainly approached using deep learning methods. For instance, Lee et al. [8] and Huynh et al. [6] use convolutional neural networks to classify tweets into those containing ADEs and those not containing ADEs. Instead of training them in a supervised way, the authors propose a semi-supervised approach to deal with the problem of scarce labeled data. Similar to our method, the predictions of different models are combined in an ensemble and a result is obtained using majority voting.

Another semi-supervised model was proposed by Gupta et al. [5]. The authors perform binary ADE classification on tweets with a bidirectional LSTM-based model. Before being fine-tuned in a supervised way, the model is pre-trained on unlabeled data for the task of predicting a drug name from its context in the tweet. More recent research investigates ADE detection as a multi-label classification task, similar to our approach. Chaichulee et al. [2] work with a dataset of free-text allergy records from electronic health records. Just like the MedNLP-SC shared task dataset, it is heavily unbalanced.The authors propose the AllergyRoBERTA model, which was pre-trained from scratch on drug allergy reports, and compare it to the pre-trained XLM-RoBERTa model.

Instead of addressing the scarcity of labeled data by using semi-supervised learning methods, we propose the use of the recent instruction-tuned LLM ChatGPT 3.5 to generate additional training data [11]. Similar to prior work, we also use ensemble learning methods to combine the strengths of different models effectively.

System message: 'Du bist ein Tweet Generator. Die Tweets enthalten keine Hashtags und keine Emojis. Formatiere deine Antworten immer wie folgt:\n1. "Tweet"\n2. "Tweet"\n3.'
Translation: 'You are a tweet generator. The tweets do not contain hashtags or emojis. Always format your answers as follows: \n1. "Tweet"\n2. "Tweet"\n3.'

User message: 'Generiere 20 Tweets, die alle folgende Bedingungen erfüllen: Die schreibende Person erzählt von {symptom} als Nebeneffekt eines Medikaments, das sie genommen hat. Der spezifische Name des Medikaments wird erwähnt. Dafür wird der Platzhalter [Medikamentenname] genutzt. Neben {symptom} werden keine weiteren Nebenwirkungen erwähnt. In einigen Sätzen wird erwähnt, wogegen das Medikament genommen wurde.'
Translation: 'Generate 20 tweets that meet all of the following conditions: The person writing talks about {symptom} as a side effect of a medication they have been taking. The specific name of the drug is mentioned. The placeholder [drug name] is used for this. No other side effects besides {symptom} are mentioned. In a few sentences, it is mentioned what the medication was taken for.'

Figure 1: Prompt used to generate additional tweets including English translation.

3 MATERIALS

For the challenge, we have access to four multi-label datasets in Japanese, English, French, and German. The data consists of synthetic social media texts in the style of tweets, each manually annotated with ADEs. The text was generated in Japanese and then translated into the other languages. A more detailed overview of the data can be found in the task paper [14].

The datasets are divided into a publicly available training and a held-out test set. Our access to labels is constrained to the primary training subset, encompassing 7,964 individual data instances. The corresponding held-out test set comprises 1,993 instances and will be referred to as $test_{final}$.

Exploratory data analysis reveals a heavily imbalanced class distribution. For example, the most occurring ADE is *nausea* with 925 data points, while the class *bone marrow dysfunction* occurs only nine times. In addition to class imbalance, the number of ADEs per instance also varies substantially. Approximately 68.5 % of the dataset does not have any ADEs, whereas 16.7 % have one, 11.3 % have two and only 3.5 % have three or more ADEs.

For evaluating our approaches, we organize the available training data into our own separate training, validation, and test subsets, maintaining a ratio of 70 %, 15 %, and 15 % respectively. To address the inherent imbalance among label classes, we use a stratified sampling. Furthermore, we combine all datasets to create a multilingual dataset by concatenating the different splits. To ensure comparability with monolingual datasets, we employ the same random seed for splitting each monolingual dataset. This maintains the class distribution and prevents dependencies between splits.

4 METHODS

In this section, we share details about our incorporated methodology as well as our data augmentation and ensemble learning approaches.

4.1 Base Model

Our approach is based on pre-trained RoBERTa models [9], which are fine-tuned it on language-specific datasets. For all languages except English, we use the large version of the multilingual XLM-RoBERTa [3] through the model from Transformers library [15]. This version of RoBERTa was trained on a multilingual dataset containing texts in 100 languages, including the four languages relevant for the shared task. For the English dataset, we use a standard RoBERTa model (large) pre-trained on English text only. For each language-specific dataset, we fine-tune the model for our multi-label classification task.

4.2 Data Augmentation

To address the issue of class imbalance, we generate artificial tweets similar to the ones in our dataset. To this end, we use the instruction-tuned *gpt-3.5-turbo* model through the OpenAI API [11].

Designing an appropriate prompt template was an iterative process. Our final prompt is shown in Figure 1: it is split into a system message that informs about the general behavior and style of the output and a user message that contains the concrete request. It enforces that each tweet should contain only one adverse drug event, to make labeling easier. The prompt contains a placeholder where we insert the side effect symptom that should be mentioned in the generated tweets. We include some other requirements to make the generated tweets similar to the ones from the dataset, for example, mentioning a specific drug and which disease it was taken for.

With the designed prompt, 20 tweets at a time are generated. Creating more tweets at a time often resulted in excessively repetitive content. Therefore, we prompt the model several times for each symptom. We then process the generated response by extracting the single tweets and replacing the medication placeholder with a name randomly chosen from a set of 22 medicament names. Additionally, we manually review all tweets and adjust those mentioning unwanted ADEs, remove duplicates, or include other mistakes. For example, a frequent problem are tweets with the class *pain*, mentioning *headache* or *abdominal pain*. These symptoms belong to separate classes and should therefore not be classified as pain.

In our experiments, we compare multiple options for including generating examples. Either, we add examples for all classes or for low-frequency classes only. Due to the time-consuming manual curation process, we do not generate tweets for all languages. Instead, we translate the generated and filtered German tweets with a machine translation models. To generate the English version from German, we use the pre-trained model `Helsinki-NLP/opus-mt-de-en` through the Transformers library. To translate from English to French and Japanese, we use similar models.

4.3 Ensemble Learning

Ensemble learning is a powerful way to increase accuracy and model robustness for classification tasks [4, 13, 16]. Multiple base learners are trained on a transformation of the original training data. For inference, their predictions are combined by a voting schema or mathematical function. The number of base learners (ensemble size) is a crucial hyperparameter. In our approach, we use an ensemble size of five, which is a trade-off between computational cost and performance improvement. Besides, using an odd number, there will always be a majority when voting is used. There are multiple approaches for creating ensembles. On the dataset level, transformations are applied to create different versions of the training data. We consider the following alternatives.

4.3.1 Dataset Methods. On the base learner level, we only change the random seed for every base learner. Alternative methods consider different hyperparameters for every base learner, or even different base learner architectures.

K-fold A k-fold split is created with the combined training and validation data. These new splits are then distributed to the k base learners.

Shuffle For every base learner, the combined training and validation data is shuffled randomly and split again without stratification.

4.3.2 Voting Techniques. There are also different options on the voting level, where the decision is made on how to combine the base learner predictions. Here, computing different variations is relatively cheap—therefore, we compute them all and choose the ones that perform best.

MajorityVote (MV) The label is true when the majority of base learners decide on this label.

AvgProbVote (APV) The average is computed out of the base learners' probability outputs. The label is true when the average is greater or equal to the decision threshold.

MaxProbVote (MaxPV) The maximum is taken out of the base learners' probability outputs. Label is true when the maximum is greater or equal to the decision threshold.

MedianProbVote (MedPV) The median is computed out of the base learners' probability outputs. The label is true when the median is greater or equal to the decision threshold.

WeightVote (WV) A weight from 1 to k is selected for every base learner, resulting from the micro F_1 score on the validation split. The label is true when the sum of the weighted base learner predictions is greater or equal to the sum of 1 to k divided by 2.

4.3.3 Further Approaches. Furthermore, we evaluate the impact of augmentation on ensemble learning. To this end, we apply the best selected augmentation method to the training data before using the corresponding dataset-level method. Moreover, we test the difference between monolingual and multilingual ensembles by building an ensemble with the multilingual dataset, combined from all language-specific subsets.

4.4 Metrics and Model Selection

For selecting the best combinations of ensemble learning and augmentation methods, we evaluate mainly on the German data. The main metric for our evaluation is the F_1 score, micro-averaged across all classes. This metric correlates well with the metrics relevant for the shared task evaluation [14]. To determine the impact of the augmentation on class imbalance, we also report the macro-averaged F_1 scores for these experiments, as it better reflects improvements for rare classes.

First, we perform an extensive hyperparameter sweep for model selection. Therefore, we use the Weights and Biases platform with a Bayesian search strategy, maximizing the micro F_1 score to select the optimal hyperparameters for the base model [1]. We select the following hyperparameters for the sweep: batch size, learning rate, number of training epochs, decision threshold, and weight decay. Preliminary experiments suggested that those hyperparameters have the most substantial impact. For the other hyperparameters, we simply use the default values of the Transformers library [15].

The following hyperparameters resulted in the best performance on our internal test set:

- Batch size of 64,
- Learning rate of 2.3635×10^{-5},
- Training epochs of 20,
- Decision threshold set at 0.5, and
- Weight decay of 0.018232.

After setting the hyperparameters for the base models, we conduct experiments to select the best augmentation and ensemble methods. Finally, we select the three best runs for the shared task submission. By selecting a different method for each run, we have the chance to compare different approaches as follows.

(1) We use a pure base model, fine-tuned with the best augmentation method for our first submission for each language.

(2) The second submission is an ensemble obtained with the respective, best ensemble learning method. Thus, we submit the best augmentation and best ensemble learning method found for each language.

(3) The final submission is either based on a multilingual ensemble, or an ensemble with augmented training data, depending on which run performs better on our internal test data. Thereby, we can also evaluate the combination of ensemble learning with either augmentation or multilingual data.

5 EXPERIMENTS

In this section, we share details about our experimental setup for data augmentation and ensemble learning. We also introduce the approaches that we evaluate on the held-out test set ($test_{final}$).

5.1 Data Augmentation

We compare different runs to evaluate our data augmentation approach. We generate 60 examples for each class, except for *pain*, for which we generate 68 examples. For the "all" setting, we add the generated examples for all classes. For other runs, we only include generated examples for selected classes. Here, we select classes for which the model performed badly in the baseline run, or for which we do not have many examples in the dataset. We evaluate approaches using different subsets of classes. In the "3-classes"-run, we add generated examples for the classes *bone marrow dysfunction*, *interstitial lung disease*, and *pain*. The "6-classes" run additionally includes examples for *hemorrhagic cystitis*, *renal impairment*, and *liver damage*.

5.2 Ensemble Learning

To find the best ensemble methods, we first create ensembles with the two dataset-level methods "K-fold" and "Shuffle". Those dataset methods make use of either the default train data, the train data with the best-chosen augmentation method, or our multilingual dataset. Every base learner is trained on the transformed data separately. After training, we compute micro F_1 scores on the internal test data with all voting techniques, selecting the best two approaches for the submission as described in the previous section. Furthermore, we compare the performance of the best base learners with the ensemble results, to determine whether the ensemble improve performance.

Table 1: Results of Augmentation Experiments on the German Test Set

	Baseline	All	6-classes	3-classes
Micro F_1 score	.830	.819	.833	**.840**
Macro F_1 score	.758	.715	**.782**	.773

5.3 Submission

For each language, three runs are submitted with predictions for the $test_{final}$ dataset. We always submit one augmentation run (submission 1), using generated examples for three classes, as explained in subsection 4.2. Additonally, we submit the following runs, selected according to their performance on our internal test set:

- For the German dataset, we use an ensemble using the K-fold dataset technique and MajorityVote (submission 2), as well as an ensemble trained on the multilingual dataset with K-fold and AvgProbVote (submission 3),
- For the English dataset, we submit an ensemble with K-fold dataset re-combinations and MajorityVote (submission 2), as well as an ensemble with K-fold and AvgProbVote with Augmentation (submission 3),
- The ensemble submissions for Japanese are a combination of Shuffle and AvgProbVote (submission 2) and an ensemble with Shuffle and WeightVote and data augmentation (submission 3), and
- For the French dataset, we include an ensemble with Shuffle and AvgProbVote (submission 2) and an ensemble trained on the multilingual dataset with K-fold and AvgProbVote (submission 3).

We compare these runs to the best performing model $XLM - R_{all}$ from the task paper [14], which is an XLM-RoBERTa model fine-tuned on the combined dataset of all four languages.

6 RESULTS

In this section, we share our internal experimental results and ablation experiments on the German dataset. Furthermore, We present the results for all languages and runs in the final submission.

6.1 Data Augmentation

Table 1 shows the results for different choices of classes subject to data augmentation. We can see that the run using generated examples for all classes falls approx. one to four pp behind the baseline results. However, adding examples for only some classes performs better. For example, the "6-classes" run achieves slightly better scores with 0.3 pp compared to the baseline for the micro F_1 score and clearly exceeds the baseline with 2 pp regarding the macro F_1 score. The best results are obtained using the "3-classes" run, with better results than the baseline for both F_1 scores and the best overall results for the micro F_1 score.

Table 2: Selected Individual Symptom F_1 scores for Augmentation Experiments on German Test Set

	Baseline	6-classes	3-classes
Interst. Lung Disease	.500	**.800**	**.800**
Bone Marrow Dysf.	.000	**.667**	.000
Pain	.468	.553	**.617**
Hemorrhagic Cystitis	**1.000**	**1.000**	.667
Renal Impairment	.500	**.545**	.444
Liver Damage	**.667**	.571	.571

Table 3: Micro F_1 scores for Ensemble Learning Experiments on German Test Set. For every ensemble, the best base learner (BL) is compared with the results of the voting techniques. The underlined values are the ones submitted for the challenge.

	BL	MV	APV	MaxPV	MedPV	WV
Shuffle	.836	.835	**.838**	.831	.835	.833
K-fold	.841	**.846**	.843	.830	**.846**	.844
Shuffle+Aug	.837	.834	**.838**	.827	.834	.830
K-fold+Aug	.823	.830	**.831**	.819	.830	.826
Shuffle+Multi	.841	**.844**	.843	.830	**.844**	.841
K-fold+Multi	.838	.846	**.849**	.823	.846	.844

We also compute individual F_1 scores for the classes that we add generated examples for in the "6-classes" run, which are depicted in Table 2. The first three rows include the classes that we used in the "3-classes" run. For two of these classes, the F_1 scores surpass baseline results. For the "6-classes" run, F_1 scores for all classes except *liver damage* and hemorrhagic cystitis are improved.

6.2 Ensemble Learning

Table 3 shows the micro-averaged F_1 scores achieved by the ensembles on our German test data. We can see there that the best base learner always performs worse than the best ensemble voting method. Considering the dataset techniques, the K-fold ensemble performs better on the pure German data and multilingual data compared to the shuffling, except when MaxProbVote voting is used.

Regarding the different datasets, the augmented data delivers the worst results, and the ensembles trained on the multilingual dataset perform best. In combination with K-fold, the ensemble trained on the pure German also performs well. The two best runs overall are the pure German K-fold ensemble with MajorityVote and the Multilingual K-fold ensemble with AvgProbVote.

6.3 Final Submission Results

Table 4 summarizes our results on the $test_{final}$ dataset. For all our submission results, we outperform the baseline in across all metrics. This confirms the results we obtained using our internal development test set. The difference to the baseline is especially high on the German dataset. Here, the per-class F_1 score (micro averaged) is improved by 5 pp compared to the baseline in the augmentation run and by 7 pp in the ensemble runs. The performance improvements compared to the baseline are also evident for all other languages. For most metrics, the augmentation approach is in third place, and one of the ensemble submissions comes out on top. The results for the two ensembles are always quite similar. Often, none of them is consistently better in all metrics.

Besides these differences between our approaches, we also see differences between languages. On the Japanese dataset, we get the best results, whereas the German and French datasets have the lowest scores. The highest overall scores are achieved by the first Japanese ensemble without augmentation. The exact accuracy reaches 0.87, and the per-class F_1 score 0.82. Between the different languages, the best accuracy ranges from 0.84 to 0.87, and the per-class F_1 scores from 0.77 to 0.82.

7 DISCUSSION

In the following, we discuss our experiments and the obtained results of the task. Additionally, we address limitations of our work.

7.1 Impact of Data Augmentation

Our experiments indicate that adding generated data for a small number of classes might support the model to discriminate these classes better, resulting in overall increased performance. Especially when analyzing the F_1 scores for the individual classes in Table 2, we notice that for the "3-classes" run the goal of improving the predictions for the selected classes was achieved in two of three cases. This is the desired result because it means that the model successfully learned from the additional generated examples. Still, we must evaluate these scores critically. Especially for *interstitial lung disease* and *bone marrow dysfunction*, our test dataset contains only a few examples. Therefore, the meaningfulness of F_1 scores for these classes are limited.

We find that adding generated data only improves performance when performed for few classes. When adding data for more classes, performance decreases. The suspected reason could be a decrease in the diversity of training examples, as the generated tweets tend to look very similar. Through more careful prompt design and optimization of the tweet generation process, we might obtain more diverse generated samples. Then, we could add more of them to our dataset to further improve the classification results.

In the "6-classes" run, we can see that even though the micro F_1 score does not significantly outperform the baseline, the macro F_1 score is the highest of all runs. We expect that in this case, the small classes for which examples are added are improved and other classes with many examples get worse due to less diversity in the sentences. The improvement of small classes can be verified with the individual F_1 scores in Table 2. When interpreting the results, we need to consider that the tweets we generated only contain one ADE per tweet. Hence, adding too many of them could also negatively affect the model's ability to predict of multiple ADEs in one tweet. It is likely that adding generated tweets with several ADEs per tweet would further improve the results.

7.2 Impact of Ensemble Learning

We outlined in subsection 6.2 that the best base learner performs always worse than the best ensemble learning. Besides that, we see

Table 4: Final submission results evaluated on $test_{final}$. For all submitted runs, we report exact accuracy (Acc.), binary F_1 score (Bin. F_1), per label F_1 score (Label F_1), and the micro averaged F_1 score per class. As a baseline, we refer to the XLM-R_{all} run provided by the MedNLP-SC organizers [14].

#	Language / Submission	Acc.	Class	Bin. F_1	Label F_1	Class F_1
	German					
	MedNLP-SC Baseline	.80	non-ADE	.89	.99	.71
			ADE	.74	.71	
1	German Data + Augmention	.83	non-ADE	.90	1.00	.76
			ADE	.77	.76	
2	German Data + Ensemble (K-fold + MV)	.84	non-ADE	.91	1.00	.78
			ADE	.78	.78	
3	Multilingual Data + Ensemble (K-fold + APV)	**.85**	non-ADE	**.92**	**1.00**	**.78**
			ADE	**.80**	**.78**	
	English					
	MedNLP-SC Baseline	.83	non-ADE	.91	.99	.76
			ADE	.78	.76	
1	English Data + Augmentation	**.85**	non-ADE	.91	1.00	.78
			ADE	.80	.78	
2	English Data + Ensemble (K-fold + MV)	.84	non-ADE	**.92**	**1.00**	**.79**
			ADE	**.81**	**.79**	
3	English Data + Augmentation + Ensemble (K-fold + APV)	**.85**	non-ADE	**.92**	**1.00**	**.79**
			ADE	**.81**	**.79**	
	Japanese					
	MedNLP-SC Baseline	.84	non-ADE	.91	1.00	.77
			ADE	.79	.77	
1	Japanese Data + Augmentation	.85	non-ADE	.91	1.00	.80
			ADE	.80	.80	
2	Japanese Data + Ensemble (Shuffle + APV)	**.87**	non-ADE	**.92**	**1.00**	**.82**
			ADE	**.82**	**.82**	
3	Japanese Data + Augmentation + Ensemble (Shuffle + WV)	.86	non-ADE	**.92**	1.00	.81
			ADE	**.82**	.81	
	French					
	MedNLP-SC Baseline	.81	non-ADE	.90	.99	.73
			ADE	.76	.73	
1	French Data + Augmentation	.83	non-ADE	.91	.99	.75
			ADE	.78	.75	
2	French Data + Ensemble (Shuffle + APV)	**.84**	non-ADE	**.92**	**1.00**	**.77**
			ADE	**.80**	**.77**	
3	Multilingual Data + Ensemble (K-fold+APV)	**.84**	non-ADE	.91	**1.00**	**.77**
			ADE	.79	**.77**	

in the final submission Table 4 that every ensemble performs better than the methods without ensemble methods (augmentation and baseline). This finding highlights that ensemble learning indeed improves the prediction performance of the classification task, as expected.

A major learning comes from choosing the underlying dataset. Overall, the best results in the competition come from the combination of multilingual or augmented data with ensemble techniques. In the German and French dataset, the multilingual ensembles achieve the best results. This is opposed to the situation for the English and Japanese dataset, where the augmented ensembles perform better than the multilingual ensembles. We assume that multilingual ensembles just perform better on more difficult tasks. Detailed results can be found in Appendix A.

We suspect that the quality of the translated German and French data might be lower compared to Japanese and English. Overall, the German and French have lower results compared to English and Japanese. Nevertheless, both multilingual and augmented ensembles deliver comparable good results. However, the improvements are not consistently observed or large enough to derive specific conclusions from them. Due to time constraints, we could not evaluate the combination of augmentation and training on the multilingual dataset. Such a combination might improve the evaluation results even further.

7.3 Submission Performance

The final submission results show that our approaches generalize well and produce competitive results even on unseen data. The substantial difference between our scores and the baseline shows that both augmentation and ensemble learning both are valuable additions to the baseline approach of using only a standard pretrained RoBERTa model. It is not surprising that we can see an especially large difference to the baseline on the German dataset, as this is the data we used to select and tune our methods.

The combination of augmentation and ensemble learning was only used in a subset of submissions, so we cannot finally evaluate how well this approach works in general. Our results suggest that the impact of augmentation is rather small, compared to using standard ensemble learning. The same is true for the training on multilingual datasets. These combinations could still be investigated more intensively, but were out of the scope of this work.

7.4 Limitations

While there is potential for improvement as touched above, we suspect that the achievable performance is limited by the quality of the dataset. Some generated tweets do not seem plausible, which might be caused by the generative model or the translation process. During manual inspection of the data, we noticed instances that pose challenges regarding comprehension or raise suspicions regarding the accuracy of their labeling. An example of it is depicted in Figure 2. Here it is difficult to understand whether pain is a side effect or not.

Text: I had my second steroid shot this week and it seems more painful than last time. It's a little better than yesterday, but... I was applying it before I went to bed today like crazy, so I guess it's not working <url>
Label: None

Figure 2: Example of an ambiguous tweet, where it is unclear whether pain is an ADE or not.

8 CONCLUSION

In this work, we presented our contribution to the NTCIR-17 SMADE subtask. We achieved very good performance results by fine-tuning RoBERTa models on monolingual and multilingual datasets in combination with data augmentation and ensemble learning methods. By using fine-tuned combinations of both techniques, we have achieved the second-best results across all languages in the main challenge metrics. Having said this, we still believe that our approach can be improved further, e.g., by generating examples with multiple positive labels or combining the multilingual ensembles with data augmentation. Furthermore, a more comprehensive analysis of different base learner variations is warranted. Nevertheless, our approach generalizes well to the held-out test data and is also easy to adapt for application on similar NLP tasks.

ACKNOWLEDGEMENTS

Parts of this work were generously supported by a grant of the German Federal Ministry of Research and Education (01ZZ2314N).

REFERENCES

[1] Lukas Biewald. 2020. Experiment Tracking with Weights and Biases. https://www.wandb.com/ (Accessed: Oct 25th, 2023).

[2] Sitthichok Chaichulee, Chissanupong Promchai, Tanyamai Kaewkomon, Chanon Kongkamol, Thammasin Ingviya, and Pasuree Sangsupawanich. 2022. Multi-label Classification of Symptom Terms from Free-text Bilingual Adverse Drug Reaction Reports Using Natural Language Processing. *PLOS ONE* 17, 8 (08 2022), 1–22. https://doi.org/10.1371/journal.pone.0270595

[3] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

[4] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on Ensemble Learning. *Frontiers of Computer Science* 14, 2 (2020), 241–258. https://doi.org/10.1007/s11704-019-8208-z

[5] Shashank Gupta, Sachin Pawar, Nitin Ramrakhiyani, Girish Keshav Palshikar, and Vasudeva Varma. 2018. Semi-supervised Recurrent Neural Network for Adverse Drug Reaction Mention Extraction. *BMC Bioinformatics* 19 (2018), 1–7. https://doi.org/10.1186/s12859-018-2192-4

[6] Trung Huynh, Yulan He, Alistair Willis, and Stefan Rueger. 2016. Adverse Drug Reaction Classification With Deep Neural Networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, 877–887.

[7] Jérémy Lardon, Redhouane Abdellaoui, Florelle Bellet, Hadyl Asfari, Julien Souvignet, Nathalie Texier, Marie-Noëlle Beyens, Anita Burgun, and Cedric Bousquet. 2015. Adverse Drug Reaction Identification and Extraction in Social Media: A Scoping Review. *Journal of medical Internet research* 17 (07 2015), e171. https://doi.org/10.2196/jmir.4304

[8] Kathy Lee, Ashequl Qadir, Sadid A. Hasan, Vivek Datla, Aaditya Prakash, Joey Liu, and Oladimeji Farri. 2017. Adverse Drug Event Detection in Tweets with Semi-Supervised Convolutional Neural Networks. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Switzerland, 705–714. https://doi.org/10.1145/3038912.3052671

[9] 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. *arXiv* abs/1907.11692 (2019). https://doi.org/10.48550/arXiv.1907.11692

[10] Rachel M. Murphy, Joanna E. Klopotowska, Nicolette F. de Keizer, Kitty J. Jager, Jan Hendrik Leopold, Dave A. Dongelmans, Ameen Abu-Hanna, and Martijn C. Schut. 2023. Adverse Drug Event Detection Using Natural Language Processing: A Scoping Review of Supervised Learning Methods. *PLOS ONE* 18, 1 (01 2023), 1–26. https://doi.org/10.1371/journal.pone.0279842

[11] OpenAI. 2023. ChatGPT 3.5. https://platform.openai.com/docs/models/gpt-3-5. (Accessed: Oct 25th, 2023).

[12] Lisa Raithel, Philippe Thomas, Roland Roller, Oliver Sapina, Sebastian Möller, and Pierre Zweigenbaum. 2022. Cross-lingual Approaches for the Detection of Adverse Drug Reactions in German from a Patient's Perspective. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 3637–3649. https://aclanthology.org/2022.lrec-1.388

[13] Omer Sagi and Lior Rokach. 2018. Ensemble Learning: A Survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1249. https://doi.org/10.1002/widm.1249

[14] Shoko Wakamiya, Lis Kanashiro Pereira, Lisa Raithel, Hui-Syuan Yeh, Peitao Han, Seiji Shimizu, Tomohiro Nishiyama, Gabriel Herman Bernardim Andrade, Noriki Nishida, Hiroki Teranishi, Narumi Tokunaga, Philippe Thomas, Roland Roller, Pierre Zweigenbaum, Yuji Matsumoto, Akiko Aizawa, Sebastian Möller, Cyril Grouin, Thomas Lavergne, Aurélie Névéol, Patrick Paroubek, Shuntaro Yada, and Eiji Aramaki. 2023. NTCIR-17 MedNLP-SC Social Media Adverse Drug Event Detection: Subtask Overview. In *Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR-17)*. National Institute of Informatics (NII), Tokyo, Japan. https://doi.org/10.20736/0002001327

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[16] Yongquan Yang, Haijun Lv, and Ning Chen. 2023. A survey on Ensemble Learning Under the Era of Deep Learning. *Artificial Intelligence Review* 56, 6 (2023), 5545–5589.

A APPENDIX

Table 5: Micro F1 Scores for Ensemble Learning Experiments on French Test Set as described above.

	BL	MV	APV	MaxPV	MedPV	WV
Shuffle	.842	.853	**.856**	.819	.853	.848
K-fold	.837	.841	**.844**	.825	.841	.840
Shuffle+Aug	.843	.847	**.852**	.814	.847	.851
K-fold+Aug	.844	.842	**.846**	.821	.842	.839
Shuffle+Multi	.843	**.844**	**.844**	.832	**.844**	.843
K-fold+Multi	.852	.863	**.864**	.841	.863	.859

Table 6: Micro F1 Scores for Ensemble Learning Experiments on English Test Set as described above.

	BL	MV	APV	MaxPV	MedPV	WV
Shuffle	.862	.861	.858	.842	.861	**.866**
K-fold	.857	**.869**	.868	.842	**.869**	**.869**
Shuffle+Aug	.861	.861	**.866**	.849	.861	.860
K-fold+Aug	.866	.872	**.873**	.854	.872	.871
Shuffle+Multi	.858	**.862**	.860	.838	**.862**	.855
K-fold+Multi	.860	.863	**.865**	.840	.863	.864

Table 7: Micro F1 Scores for Ensemble Learning Experiments on Japanese Test Set as described above.

	BL	MV	APV	MaxPV	MedPV	WV
Shuffle	.880	.888	**.891**	.859	.888	.881
K-fold	.884	.886	**.888**	.868	.886	.884
Shuffle+Aug	.888	.888	.889	.871	.888	**.891**
K-fold+Aug	.881	.883	.879	.870	.883	**.886**
Shuffle+Multi	.880	**.885**	.882	.857	**.885**	.884
K-fold+Multi	.875	.877	**.879**	.860	.877	.870