Towards Efficient Dialogue Processing in the Emergency Response Domain

Tatiana Anikina
DFKI / Saarland Informatics Campus,
Saarbrücken, Germany
tatiana.anikina@dfki.de

Abstract

In this paper we describe the task of adapting NLP models to dialogue processing in the emergency response domain. Our goal is to provide a recipe for building a system that performs dialogue act classification and domain-specific slot tagging while being efficient, flexible and robust. We show that adapter models (Pfeiffer et al., 2020) perform well in the emergency response domain and benefit from additional dialogue context and speaker information. Comparing adapters to standard fine-tuned Transformer models we show that they achieve competitive results and can easily accommodate new tasks without significant memory increase since the base model can be shared between the adapters specializing on different tasks. We also address the problem of scarce annotations in the emergency response domain and evaluate different data augmentation techniques in a low-resource setting.

1 Introduction

Emergency response is a very challenging domain for NLP for a variety of reasons. First, this domain has strict requirements regarding memory and computational efficiency. Often it is not feasible to load several large NLP models because of the limitations in the available infrastructure (e.g., memory of the machine where the models are running). Second, the environment is often noisy and the speakers communicate using domain-specific lexicon and abbreviations. Third, emergency situation environment is very changeable and the domain may vary from a rescue operation in a car accident to explosions or building collapse. Hence, the ideal dialogue processing system for the emergency response domain should be memory efficient, robust and flexible at the same time.

To address the efficiency aspect we use adapters[1] (Pfeiffer et al., 2020) that were tested on a variety of NLP tasks and have shown a comparable performance with the full fine-tuning while using only 1% of the parameters of the fully fine-tuned models. Adapters are small in size, can be easily shared and combined with different models. This is especially interesting in our use case since we deploy the same base model (`bert-base-german-cased`) for several tasks[2].

To tackle the problem of noisy, incomplete and domain-specific communication we investigate whether it is possible to boost the performance by integrating additional context and experiment with different ways of encoding it (e.g., by adding speaker, previous turn and dialogue summary information). We also experiment with various linguistic features and test how they affect the performance (e.g., by embedding the POS tags or including the ISO-style dialogue act annotations).

Finally, to simulate the low-resource scenario which is very common for the emergency response domain we reduce the amount of the training and development data to 12% of the original dataset and apply different ways of data augmentation including backtranslation, LM-based word replacements and random edit operations.

Figure 1 provides an overview of different experimental settings addressed in this work. To our knowledge, this is the first work that explores dialogue processing in the emergency response domain with adapters and performs a comprehensive study of the context integration and data augmentation in this setting.

2 Related Work

Adapters (Houlsby et al., 2019; Rebuffi et al., 2017) seem like a natural choice for lightweight and ef-

[1]The code and the pre-trained models are available at https://github.com/tanikina/emergency_response_dialogue

[2]We also tried multilingual BERT but it resulted in worse performance in our pilot experiments. Hence, we decided to focus on the model that was trained on German only and has a reasonably small size (436 MB).

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics - Student Research Workshop, pages 212–225

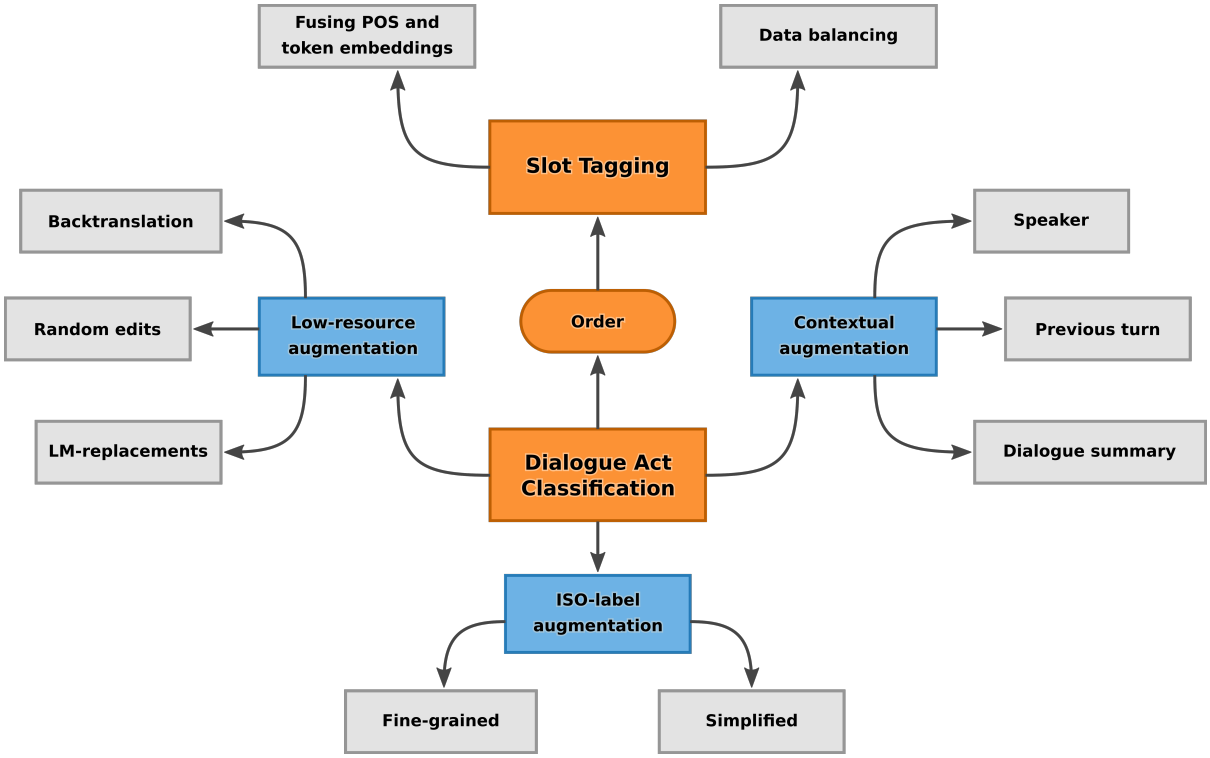

Figure 1: Overview of the Experiments

ficient NLP models. Adapters implement a fine-tuning strategy that involves only a small amount of trainable parameters per task. Each adapter adds a small set of newly initialized and trainable weights at each layer of the transformer architecture (Vaswani et al., 2017). Hence, the original network has mostly fixed parameters and can be efficiently transferred between the tasks. Adapters have shown good performance comparable to the fully fine-tuned models on a variety of tasks including, e.g., sentiment analysis, commonsense reasoning, paraphrase detection and entailment (Pfeiffer et al., 2021) and further modifications and improvements to the original idea were proposed in the recent work by Rücklé et al. (2020); Fu et al. (2022). Adapters have been successfully used for low-resource speech recognition (Hou et al., 2021), cross-lingual transfer (Parovic et al., 2022) and tested on the named entity recognition and classification tasks (Lee et al., 2022).

Also, in the field of dialogue processing there is a growing body of work involving adapter models. For example Xu et al. (2021) inject knowledge into pre-trained language models using adapters and explore grounded dialogue response generation with adapters. Another work by Madotto et al. (2020) proposes a simple and efficient method based on residual adapters in the continual learning setting for task-oriented dialogue systems. Wang et al. (2021) design a GPT-Adapter-CopyNet system that combines adapters and CopyNet modules into GPT-2 in order to perform transfer learning and dialogue entity generation. Their system significantly outperforms the baselines models on both DSTC8 and MultiWOZ data.

Efficiency and robustness are crucial in the low-resource setting when we have a limited amount of data. The main objective of data augmentation is to generate new data points by modifying the existing ones through a variety of transformations and while some of these transformations can be very simple such as random token deletion or insertion (Wei and Zou, 2019; Miao et al., 2020), others might require more computation and processing power, e.g., backtranslation (Edunov et al., 2018) or LM-based substitutions (Kobayashi, 2018; Kumar et al., 2020). Feng et al. (2021) and Chen et al. (2021) provide comprehensive surveys of the techniques and methods for data augmentation in NLP that served as a motivation for our work.

3 Data

The dataset used in our experiments is based on the dialogues collected during several robot-assisted disaster response training sessions (Kruijff-

Korbayova et al., 2015; Willms et al., 2019). All dialogues are in German and they represent team communication between a team leader or mission commander and several operators who remotely operate robots in order to explore some area, find hazardous materials, locate fires, damage or victims. Figure 2 shows a part of one dialogue translated into English.

speaker	original turn	translation
TL:	*UGV2 von Teamleader.*	*UGV2 for team leader.*
UGV:	*UGV2, kommen.*	*UGV2, coming.*
TL:	*Ja, UGV2, wir brauchen nochmal schärfere Bilder von dem Fass und der Kennzeichnung.*	*Yes, UGV2, we need again sharper pictures of the barrel and the sign.*
UGV:	*Ich habe Sie nicht verstanden, können Sie wiederholen?*	*I didn't understand you, could you repeat?*
TL:	*Ja, von dem Fass brauchen wir nochmal bessere Bilder, und auch von der Kennzeichnung.*	*Yes, we need better pictures of the barrel, and also of the sign.*

Figure 2: Example of communication between the Team Leader (TL) and the Unmanned Ground Vehicle operator (UGV).

The complete dataset contains 2,542 dialogue turns annotated with dialogue acts and domain-specific slots. For the dialogue act classification we reserve 2,261 turns for training, 281 turns for development and 283 for testing. In the low-resource setting we leave the test set unchanged but reduce the amount of the training samples to 310 (240 in training and 70 in development).

Figure 3 shows the overall distribution of different dialogue act labels in the data and Figure 6 in the appendix provides an example for each label. There are seven main labels: Call, CallResponse, InfoRequest, InfoProvide, Confirm, Disconfirm, Order and the additional label Other for the cases that do not fit in any of the main categories. The labels are derived based on the domain expertise and represent categories that are important for the emergency response domain. Part of the dataset is also annotated according to the ISO standard for dialogue act classification by Bunt et al. (2020)

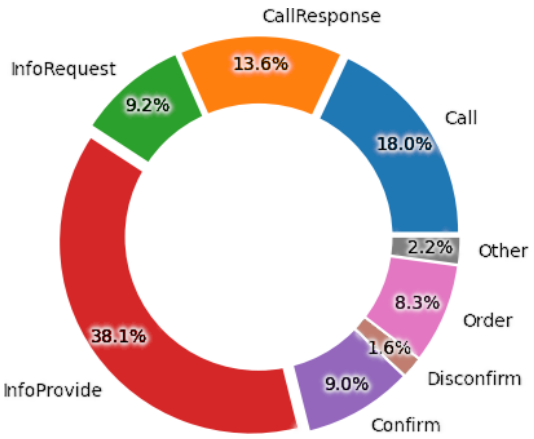

Figure 3: Dialogue Act Distribution

and we use these fine-grained labels in some of the experiments described in Section 4.

In the emergency response domain it is very important to correctly recognize and annotate all deployment orders (*Einsatzbefehl* in German). Note that not every utterance classified as request according to the ISO standard would qualify as Order in our domain. E.g., the request *"Could you repeat, please?"* is not a deployment order since it does not require performing a domain-specific action and should be classified as information request (InfoRequest).

For each turn annotated as Order we also perform the slot tagging. The slots are based on the regulation document of the emergency responders Feuerwehr-Dienstvorschrift (1999). We show an example containing all relevant Order slots in Figure 4. Note that the distribution of slots is quite uneven (see Figure 5). Some slots are present in almost every dialogue turn classified as Order (e.g., Unit is present in 67% of the turns and Task appears in 99% of them) while other slots are annotated only in 8% of the turns (Way). Also, the slots can be nested and the same token may belong to several slots. E.g., in *"Schickst du mir noch ein Foto?"* (Will you send me also the photo?), *"du"* (you) is part of the slot Task and also the slot Unit. This is the reason why we train separate models for each slot and then combine the results to provide final annotations.

For the slot tagging task we experiment with the full data as well as with the sampled data since the distribution of the negative versus positive instances per label varies a lot (see Figure 5 for the details). For the sampled data we limit the amount of negative samples (turns without the slot annota-

A-Trupp zur Brandbekämpfung mit Schaumstrahlrohr zum Pkw über die Wiese vor!
Einheit Auftrag Mittel Ziel Weg
A-Squad to extinguish the fire with a foam jet nozzle to the car across the meadow !
Unit Task Means Goal Way

Figure 4: Slot Tags for Deployment Order

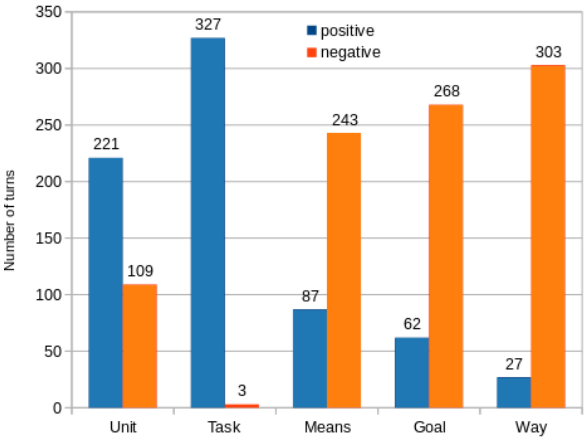

Figure 5: Slot Distribution

tion) to maximum 80% of the corresponding positive samples. Our intuition is that having uneven distribution with too many negative samples may hinder the model's performance and it might be easier for the adapter model to learn the tagging task on more balanced data. We test this idea and describe our results in the next section.

4 Experiments

Our experiments aim to answer the following research questions:

- Can we replace fully tuned BERT models with adapter models for dialogue act classification and slot tagging in the emergency response domain?
- Does integrating context and linguistic features in the model result in better performance?
- Does data augmentation in the low-resource setting help to improve the performance and what are the best ways to augment the data?

4.1 Vanilla BERT vs. Adapters

In order to check whether adapter models work well for dialogue act classification we compare their performance to vanilla BERT fine-tuned on the same data. Both models use the same base `bert-base-german-cased` model as a backbone and are trained for 20 epochs. The best performing checkpoint is selected based on the loss on the development set. When only the current turn embeddings are used as input we obtain 0.82 F1 score with the fine-tuned BERT and 0.80 F1 with the adapter model (Table 1). Adding speaker to the input results in 0.80 F1 for BERT and 0.79 F1 score for adapter.

We also compare the performance of the fully tuned BERT vs. adapters on the slot tagging task. Since the slots can be nested we train a separate model for each slot type (i.e., 5 adapters or 5 fine-tuned BERT models per setting). We use BIO notation for each slot type and compute F1 scores based on the token-level annotations. The results are summarized in Table 2. Since the distribution among the slots is uneven we also experiment with the setting where we reduce the amount of negative samples and balance the data.

It is clear from the evaluation results presented in Table 2 that adapters consistently outperform BERT on the slot tagging task and also benefit from the sampling of negative examples. Reducing the amount of negative samples gives us 9% increase in the macro F1 score for adapters while it does not bring any improvement for the vanilla BERT and effectively hurts the model's performance in terms of micro F1 (0.86 vs. 0.99). It turns out that we can use fewer parameters of the adapter model to achieve better results with the balanced classes.

Interestingly, the fully fine-tuned BERT model trained on the full data achieves the same macro F1 as the model trained on the sampled data but their micro F1 scores differ (0.99 vs. 0.86). One possible explanation is that since tuning of the BERT model involves more parameters that need to be updated in each iteration the training process becomes less stable. The difference in training stability between the adapters and the fully fledged fine-tuning in the low-resource setting is an interesting research question that needs further investigation.

Setting	Fine-tuned BERT	Adapter
OnlyTurn	0.82	0.80
Speaker+Turn	0.80	0.79
Context+Speaker+Turn	**0.91**	0.84
Context+AllSpeakers+Turn	0.90	**0.85**
Summary+Speaker+Turn	0.80	0.73

Table 1: Macro F1 scores on the dialogue act classification task (BERT vs. adapters).

Slot Label	Adapt+full	Adapt+sampled	BERT+full	BERT+sampled
Unit	0.93	0.92	0.82	0.80
Task	0.75	0.82	0.77	0.41
Means	0.86	0.89	0.82	0.88
Goal	0.57	0.81	0.59	0.67
Way	0.70	0.80	0.57	0.77
Macro F1	0.76	**0.85**	0.71	0.71
Micro F1	0.99	0.99	0.99	0.86

Table 2: Adapters (Adapt) vs. fine-tuned BERT (BERT) on the slot tagging task.

4.2 Contextual Augmentation

In the next set of experiments we look into the impact of context on the dialogue act classification (Table 1). First, we train both vanilla BERT and adapter model using only the current turn text as an input (OnlyTurn). This results in 0.82 F1 score for BERT and 0.80 F1 for the adapter. Next, we add the speaker information (Speaker+Turn) and obtain 0.80 for BERT and 0.79 for the adapter model. Moreover, adding the previous dialogue turn as additional context (Context+Speaker+Turn) results in a big improvement for both fine-tuned BERT (0.91 F1) and adapter (0.84 F1).

To integrate more context into the model input we also experiment with extractive summarization of the dialogue using the `Summarizer` model introduced in Miller (2019). We limit dialogue context to 10 previous turns and set the number of summary sentences to 3 (Summary+Speaker+Turn). However, this additional information seems to confuse the model which is especially striking in the case of adapters. Compared to the baseline Speaker+Turn (0.79 F1) the average score drops by 6 point (0.73 F1). The BERT model performance does not decrease in this setting compared to the baseline but it also does not show any improvement.

As a baseline for further experiments we use the version that encodes only the speaker information and the current turn text (Speaker+Turn). The main reason to select this setting as a baseline instead of OnlyTurn with a slightly higher macro F1 score is the fact that there is an important difference in how these two models annotate instances of the class Order. Speaker+Turn model has a better F1 score for the class Order (0.86) compared to the OnlyTurn version (0.77) and since correct processing of orders is crucial for our domain we choose this setting for the baseline. Another reason to pick Speaker+Turn and not the best-performing version that includes additional context (Context+AllSpeakers+Turn) is the fact that it is simpler and quicker to compute.

4.3 Adding Linguistic Information

Dialogue Act Classification

The subset of our dataset also provides the ISO-based annotations of dialogue acts according to Bunt et al. (2020) which we use to train a separate classifier that generates fine-grained ISO labels. These labels are added to the input of our main classifier that performs the domain-specific dialogue act classification. The distribution of the labels according to the ISO standard is shown in Table 7 in the appendix. We split the data into 1,224 samples for training and 170 for development. Although the overall accuracy of this classifier is only 62% it performs differently on different labels. The categories that have many instances in the training set (e.g., AutoPositive and TurnAccept) achieve F1 score around 0.81 and 0.82 but most of the rare labels are being misclassified.

After training the adapter-based classifier on the

ISO labels we run it on our training, development and test data to annotate the turns with additional ISO labels. Here we do not use the gold labels to simulate a realistic scenario when gold annotations are not available. The generated labels are then translated into German and added to the turn text with a special `[SEP]` token as a separator. The evaluation results are summarized in Table 3. The first column shows the scores for each of the dialogue acts when the baseline model (Speaker+Turn) is used. The second column shows the performance when additional (generated) labels are added to the input. We obtain an overall 3% improvement in the F1 scores with the additional ISO labels. We also consider a simplified version of the labels when we automatically map the original ISO taxonomy to the closest equivalents in the domain-specific taxonomy (see Table 8 in the appendix). The performance of the adapter model with such simplified dialogue act annotations is slightly worse than the ISO version (0.81 vs. 0.82).

Slot Tagging

To investigate whether linguistic annotations are also useful for the slot tagging task we annotate each word with its part of speech tag using the `SpaCy` library and 7 coarse categories including noun, pronoun, verb, preposition, adverb, adjective and other. For each tag we generate an embedding and combine it with the BERT embedding of the corresponding token. To process the combined embeddings we use a custom adapter head that adds two linear layers on top of the Transformer model, the tanh activation function and the final fully connected layer that outputs scores for the slot labels (BIO tags). The evaluation results of the adapter models with and without embedded POS information are presented in Table 4. Although the overall F1 score does not change we can see an improvement for almost every category (Task, Means and Way) except for the category Goal[3]. It is possible that for the class Goal the over-reliance on the POS information leads to some misclassifications.

4.4 Data Augmentation in the Low-Resource Setting

In order to simulate a low-resource scenario for the dialogue act classification we reduce the amount of the training and development data. The test set is left unchanged but the training set is reduced from 2,261 to 240 instances and the development set from 281 to 70 instances. As shown in Table 5 the performance drops to 0.47 F1 score on the test set when the model is trained on the reduced data.

First, we experiment with backtranslations using the `NLPAug` library. We translate between German and English and then back to German with `Helsinki-NLP/opus-mt` models and add these additional data as new instances with the same labels to the training data. This gives us an average improvement of 9 points in the F1 score. We also test whether adding more backtranslated samples helps to improve the performance and add the samples translated from German to French and back. However, doubling the amount of backtransalted data does not bring any further improvements (see Table 5). When looking at the generated backtranlations we notice that many instances are correct and represent good paraphrases. E.g., *"Und guck mal ob du ein genaues Bild von diesen Samples kriegen kannst"* (And see if you can get a clear picture of these samples) was backtranslated into *"Und sehen Sie, ob Sie ein genaues Bild von diesen Proben bekommen können"* which is semantically equivalent. However, sometimes the generated samples contain repetitions, hallucinations or incorrect translations. For example, *"Einsatzleiter"* (group leader) was translated into *"Operations Managers"* which is not a valid term in the emergency response domain.

Although backtranslation brings a substantial boost in performance, it also involves computationally heavy translation models, requires some extra processing time[4] and may not be feasible for some language pairs. Hence, we also experiment with cheaper and less time- and resource-consuming methods for data augmentation. First, we apply random masking to different proportions of the original tokens and generate substitutions using `bert-base-german-cased` language model. Table 6 shows in each row the proportion of the replaced tokens and each column shows the number of augmentation rounds. When selecting a new word for the masked token we set the parameter topk to 10 and iterate over all generated tokens to select the one that is different from the original word and does not represent a subtoken starting with ##, we also ignore all `[unused punctuation]` tokens. Some of the LM-

[3]Here we report the results of a single run but the trend was consistent among several runs of the model.

[4]It takes around 7 minutes to backtranslate 240 instances.

Dialogue Act	Adapter Baseline	Adapter+ISO DA	Adapter+simple ISO DA
Call	0.88	0.85	0.84
CallResponse	0.84	0.81	0.80
InfoRequest	0.98	0.83	0.97
InfoProvide	0.87	0.88	0.88
Confirm	0.44	0.52	0.49
Disconfirm	0.44	0.73	0.73
Order	0.86	0.83	0.79
Other	1.00	1.00	1.00
Macro F1	0.79	**0.82**	0.81

Table 3: Performance of the adapter model with and without additional ISO dialogue act labels (F1 scores).

Slot Label	Adapter Baseline	Adapter+POS
Unit	0.92	0.92
Task	0.82	0.85
Means	0.89	0.91
Goal	0.81	0.76
Way	0.80	0.82
Macro F1	**0.85**	**0.85**

Table 4: Performance of the adapters models with and without part-of-speech information on the slot tagging task.

based replacements are near-synonyms and match the context quite well (e.g., substituting *"Realbild"* (real picture) with *"Gesamtbild"* (overall picture)). However, sometimes the substituted token changes the meaning significantly. For instance, when replacing *"ja"* in *"ja kommt sofort"* (yes, coming immediately) with *"Geld"* (money) we generate a nonsensical in our domain sentence *"Geld kommt sofort"* (money comes immediately). We believe that this might be the reason why the performance of this approach is not consistently better as in case of backtranslations, although some settings (e.g., 60% LM replacements 5x) achieve similar performance. Also, we observe that replacing more than 60% tokens or augmenting more than 10 times is not beneficial for the model and leads to decreased performance.

The simplest and cheapest way of augmenting the data in terms of both time and computational resources is random editing. We add new instances by applying three different operations to randomly selected tokens: insert, delete or swap and similarly to the case of LM substitutions we experiment with different settings w.r.t. the number of edited tokens as well as the amount of the augmented data. As shown in Table 6 we get an overall improvement over the baseline model with 0.47 F1 score but there is no clear pattern regarding how many times or how many tokens should be changed. The experimental results show that the gains from adding new edited data are diminishing after 5 rounds of augmentation and the best performance can be achieved with 5 augmentation rounds and 40% edited tokens (Macro F1 0.57).

Training Details

All the experiments reported in this paper were performed on a a single GPU NVIDIA GeForce RTX 2080. We use `adapter-transformers` library to train the adapter models and `transformers` library for tuning the standard BERT models. As a base model we use `bert-base-german-cased`. We run the `SpaCy` library for the POS tag annotation with `de_core_news_sm` model for German and `Summarizer` for generating dialogue summaries. Baktranslations are performed with the data augmentation library `NLPAug`. Further details about exact versions of the software and training hyperparameters can be found in the appendix (Figures 9 and 10).

5 Discussion

Our experiments show that adapter models can be successfully applied in a very specific and challenging domain such as emergency response. Although fine-tuning BERT gives a slightly better performance (0.80 vs. 0.79 F1 for the baseline), adapters are much more efficient in terms of memory and

Dialogue Act	Baseline (full)	Baseline (low-resource)	Backtranslated 1x	Backtranslated 2x
Call	0.88	0.32	0.68	0.63
CallResponse	0.84	0.35	0.78	0.69
InfoRequest	0.98	0.87	0.70	0.79
InfoProvide	0.87	0.59	0.65	0.71
Confirm	0.44	0.56	0.66	0.65
Disconfirm	0.44	0.29	0.35	0.35
Order	0.86	0.76	0.64	0.67
Other	1.00	0.05	0.00	0.00
Macro F1	0.79	0.47	**0.56**	0.56

Table 5: Performance of the adapter model on the full and low-resource dialogue act classification with and without backtranslations (F1 scores).

LM-based word replacements				
%	**1x**	**2x**	**5x**	**10x**
0.1	0.50	0.50	0.49	0.51
0.2	0.45	0.49	0.48	0.52
0.4	0.54	0.53	0.55	0.54
0.6	0.52	0.53	**0.56**	0.54
Random edits: insert, delete, swap				
%	**1x**	**2x**	**5x**	**10x**
0.1	0.48	0.52	0.55	0.53
0.2	0.54	0.51	0.56	0.55
0.4	0.52	0.52	**0.57**	0.54
0.6	0.56	0.54	0.53	0.54

Table 6: Dialogue act classification performance (macro F1) on the augmented data. The baseline macro F1 is 0.47.

computational resources. As shown in Table 10 in the appendix an average size of an adapter model is 3.6MB compared to 436.4MB of the fully tuned BERT model. Also, adapters are very flexible and can be easily combined and stacked in different ways to perform a variety of annotations on top of the same base model.

We found that contextual augmentation (Context+AllSpeakers+Turn setting) is very beneficial for adapters and helps to increase F1 score up to 6 points compared to the baseline version. However, including longer context and dialogue summary actually confuses the model and hurts the performance. Hence, we conclude that for the dialogue act classification task the best way of integrating context is to combine the current and the previous turn with the speaker information. Adding linguistic features such as ISO dialogue acts and POS tags also helps to boost the performance but to a smaller extent (e.g. adding an ISO label increases F1 score by up to 3 points). The slot tagging task with adapters outperforms vanilla BERT in all settings and greatly benefits from the data balancing and negative sampling.

In the low-resource setting with 12% of the original data we find that adding backtranslated samples helps to improve the performance by up to 9 F1 points. However, multiple backtranslations are not necessarily useful and performance plateaus after one round of augmentation. LM-base word replacements and random edits can achieve similar performance but have a greater variance across the settings with different number of edits and augmentation rounds.

The dialogue turn tokens have different relevance to the task in the emergency response domain and replacing words blindly may result in unrealistic or simply wrong instances. E.g., *"kommen"* (coming) has a specific meaning according to the communication protocol used by the responders and represents an instance of the CallResponse class. Replacing *"kommen"* with *"gehen"* (going) or another similar verb results in the wrong interpretation and should not be labeled as CallResponse. In the future we would like to explore various constraints on the token substitutions and include more

domain knowledge and ontology information to perform targeted replacements and edits.

Active learning for text classification (Schröder and Niekler, 2020; Zhang et al., 2022) is another approach that may work well in our domain. We have already shown that adapters benefit from balancing the data and it would be interesting to see whether they further improve by learning in stages when the model starts with he balanced dataset with easy-to-classify labels and the difficulty level gradually increases with each epoch. Also, in the future we would like to explore conditional text generation with the models like BART (Lewis et al., 2019) or T5 (Raffel et al., 2020) which can be trained to generate text given the corresponding label.

6 Limitations

The main limitation of our work is the focus on the specific domain and the dataset that is not yet publicly available. However, we should note that the dataset can be requested for further research and replication studies and it will be released in the future.We believe that testing adapters with different settings in the emergency response domain is a valuable contribution but we are also aware of the fact that the dataset used in our experiments is not large or exhaustive enough to cover all the variety of topics relevant for the emergency response. For example, our data cover cases of explosions, leakages of hazardous materials and building collapse but do not include any dialogues for open field rescue operations or car accidents.

Another issue that is worth mentioning is the fact that all recordings were collected during the training sessions and not the actual missions. Hence, the responders might be under less pressure than in a real life-threatening situation and their communication might be more of a textbook case. However, all simulations had a realistic setting that includes several operators, robots and points of interest (objects or locations) and we believe that the recorded communication is representative for the domain in question.

7 Conclusion

In this work we evaluate the performance of several adapter models in the emergency response domain. We demonstrate that adapters show similar performance to the vanilla fine-tuned BERT in the baseline setting (0.79 vs. 0.80 F1 score) while using only 1% of the parameters of the fully tuned model. Our experiments show that including additional context such as previous turn and speaker can improve the performance by up to 6 points in F1 score. Also adding linguistic annotations such as ISO dialogue acts boosts the performance in dialogue act classification. The slot tagging task mostly benefits from the balanced data. As for the low-resource setting, it shows a substantial improvement over the baseline (9 F1 points) when a single round of backtranslated turns is added to the training set.

Acknowledgements

The author was supported by the German Ministry of Education and Research (BMBF) in the project CORA4NLP (grant Nr. 01IW20010).

We also thank the anonymous reviewers for their valuable feedback as well as Prof. Josef van Genabith, Dr. Simon Ostermann and Bernd Kiefer for their advice and support of this project.

References

Harry Bunt, Volha Petukhova, Emer Gilmartin, Catherine Pelachaud, Alex Chengyu Fang, Simon Keizer, and Laurent Prévot. 2020. The ISO standard for dialogue act annotation, second edition. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 549–558.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2021. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Feuerwehr-Dienstvorschrift. 1999. Feuerwehrdienstvorschrift 100 führung und leitung im einsatz: Führungssystem, bundesamt für bevölkerungsschutz und katastrophenhilfe.

Chin-Lun Fu, Zih-Ching Chen, Yun-Ru Lee, and Hung-yi Lee. 2022. AdapterBias: Parameter-efficient token-dependent representation shift for adapters in

NLP tasks. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2608–2621, Seattle, United States. Association for Computational Linguistics.

Wenxin Hou, Hanlin Zhu, Yidong Wang, Jindong Wang, Tao Qin, Renjun Xu, and Takahiro Shinozaki. 2021. Exploiting adapters for cross-lingual low-resource speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:317–329.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*.

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.

Ivana Kruijff-Korbayova, Francis Colas, Mario Gianni, Fiora Pirri, Joachim Greeff, Koen Hindriks, Mark Neerincx, Petter Ogren, Tomáš Svoboda, and Rainer Worst. 2015. Tradr project: Long-term human-robot teaming for robot assisted disaster response. *KI - Künstliche Intelligenz*, 29.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.

Jaeseong Lee, Seung-won Hwang, and Taesup Kim. 2022. FAD-X: Fusing adapters for cross-lingual transfer to low-resource languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 57–64, Online only. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Edward Ma. 2019. Nlp augmentation.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul A. Crook, Bing Liu, Zhou Yu, Eunjoon Cho, and Zhiguang Wang. 2020. Continual learning in task-oriented dialogue systems. In *Conference on Empirical Methods in Natural Language Processing*.

Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang Chiew Tan. 2020. Snippext: Semi-supervised opinion mining with augmented data. *Proceedings of The Web Conference 2020*.

Derek Miller. 2019. Leveraging BERT for extractive text summarization on lectures. *CoRR*, abs/1906.04165.

Marinela Parovic, Goran Glavas, Ivan Vulic, and Anna Korhonen. 2022. Bad-x: Bilingual adapters improve zero-shot cross-lingual transfer. In *North American Chapter of the Association for Computational Linguistics*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *NIPS*.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2020. Adapterdrop: On the efficiency of adapters in transformers. In *Conference on Empirical Methods in Natural Language Processing*.

Christopher Schröder and Andreas Niekler. 2020. A survey of active learning for text classification using deep neural networks. *ArXiv*, abs/2008.07267.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Weizhi Wang, Zhirui Zhang, Junliang Guo, Yinpei Dai, Boxing Chen, and Weihua Luo. 2021. Task-oriented dialogue system as natural language generation. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Conference on Empirical Methods in Natural Language Processing*.

Christian Willms, Constantin Houy, Jana-Rebecca Rehse, Peter Fettke, and Ivana Kruijff-Korbayová. 2019. Team communication processing and process analytics for supporting robot-assisted emergency response. In *IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2019, Würzburg, Germany, September 2-4, 2019*, pages 216–221.

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*, pages 38–45, Online. Association for Computational Linguistics.

Yan Xu, Etsuko Ishii, Zihan Liu, Genta Indra Winata, Dan Su, Andrea Madotto, and Pascale Fung. 2021. Retrieval-free knowledge-grounded dialogue response generation with adapters. In *Workshop on Document-grounded Dialogue and Conversational Question Answering*.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. A survey of active learning for natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Appendix

label	original	translation
Call	*UGV 2 von Teamleader.*	*UGV 2 for team leader.*
CallResponse	*UGV 2, kommen.*	*UGV 2, coming.*
InfoRequest	*Du sprachst eben von einer anderen Ebene, habt ihr die schon erreicht?*	*You were talking about another floor, have you already reached it?*
InfoProvide	*Foto ist erstellt und geteilt.*	*Photo was made and shared.*
Confirm	*Ja, mache ich.*	*Yes, I will do this.*
Disconfirm	*Wir haben aktuell immer noch Probleme mit der Steuerung.*	*We are currently still having problems with the controls.*
Order	*Schickst du mir noch mal ein aktuelles Foto euren Standortes?*	*Will you send me again the current photo of you position?*

Figure 6: Dialogue Act Examples

ISO Dialogue Act	Samples	ISO Dialogue Act	Samples
Allo-positive	4	Agreement	5
Auto-negative	5	DeclineOffer	5
AddressRequest	10	ChoiceQuestion	10
Instruct	10	SetQuestion	11
Pausing	17	Promise	18
AcceptOffer	19	CheckQuestion	20
TurnTake	20	Disconfirm	24
Other	29	Question	36
Confirm	37	PropositionalQuestion	38
Offer	39	Answer	45
AcceptRequest	47	Request	107
Auto-positive	159	TurnAccept	207
TurnAssign	217	Inform	255

Table 7: Distribution of the ISO dialogue acts.

Simplified Dialogue Act	Original ISO Labels
Call	TurnTake, TurnAssign
CallResponse	TurnAccept
InfoRequest	Question, ChoiceQuestion, SetQuestion, CheckQuestion, PropositionalQuestion
InfoProvide	Answer, Inform, Offer, Promise, AddressRequest, Instruct
Confirm	Confirm, Agreement, AcceptOffer, AcceptRequest
Disconfirm	Disconfirm, Auto-negative
Order	Request
Other	All other labels

Table 8: Mapping between the ISO labels and the domain-specific dialogue acts.

Library	Version	URL	Reference
Adapter-transformers	3.1.0	https://github.com/adapter-hub/adapter-transformers	Pfeiffer et al. (2020)
Transformers	4.18.0	https://github.com/huggingface/transformers/	Wolf et al. (2020)
Summarizer	0.10.1	https://github.com/dmmiller612/bert-extractive-summarizer	Miller (2019)
NLPAug	1.1.10	https://github.com/makcedward/nlpaug	Ma (2019)
SpaCy	3.2.4	https://spacy.io/	NA

Table 9: External libraries used in the experiments.

Parameters	Adapt Dialogue Acts	BERT Dialogue Acts	Adapt Slots	BERT Slots
Base Model	bert-base-german-cased		bert-base-german-cased	
Learning Rate	1e-4	1e-4	1e-3	1e-5
Number of Epochs	20	20	12	12
Batch Size	32	16	16	16
Optimizer	AdamW	AdamW	AdamW	AdamW
Avg. Training Time	6 min	22 min	4 min	4 min
Avg. Model Size	3.6MB	436.4MB	3.6MB	434.1MB

Table 10: Training parameters for different model types. The best performing model was selected based on the loss on the development set.