HHU at SemEval-2023 Task 3: An Adapter-based Approach for News Genre Classification

Fabian Billert and **Stefan Conrad**
Heinrich-Heine University of Düsseldorf
{fabian.billert, stefan.conrad}@hhu.de

Abstract

This paper describes our approach for Subtask 1 of Task 3 at SemEval-2023. In this subtask, task participants were asked to classify multilingual news articles for one of three classes: Reporting, Opinion Piece or Satire. By training an AdapterFusion layer composing the task-adapters from different languages, we successfully combine the language-exclusive knowledge and show that this improves the results in nearly all cases, including in zero-shot scenarios.

1 Introduction

In SemEval-2023 Task-3 (Piskorski et al., 2023), the authors intend to find out what makes a news article persuasive. To this end, they designed three subtasks: News Genre Categorisation, Framing Detection and Persuasion Techniques Detection. In this paper, we present our results for the first subtask, which aims to classify the genre of a news article into one of three classes; reporting, opinion piece or satire. The news articles are available in different languages, meaning this is a multilingual problem. Furthermore there are a few languages without training data in order to test the performance of the models in a zero-shot scenario. The results for the different languages were evaluated separately while taking the F1-macro score as the evaluation metric.

In the following, we present our approach for English, Italian, German and Russian, as well as the zero-shot languages Spanish and Greek. We use an adapter-based setup (Pfeiffer et al., 2020b), which enhances BERT-type architectures (Devlin et al., 2018) with modular weights. We train a task-adapter for each of the languages before using AdapterFusion (Pfeiffer et al., 2020a) to combine what the language-specific task adapters learned. We show that using AdapterFusion largely improves the performances of the single-language task-adapters, which is a behavior that has already been observed for low-resource tasks. With an F_1-macro score of 0.5942, we achieved 4th place out of 27 teams for the English dataset. In addition, for the zero-shot language Greek, we came in third place out of 15 teams with an F_1-macro score of 0.7504. For the other languages, we ranked near 10th place, meaning that our approach still shows some instabilities.

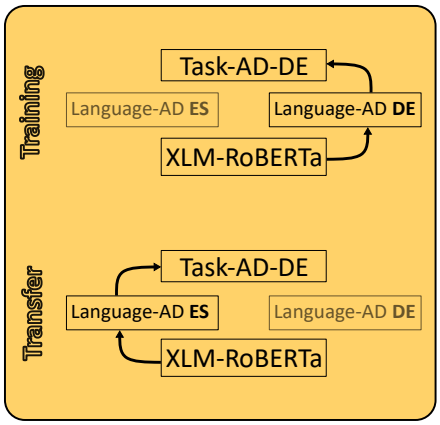
2 Background

The trainig data for subtask-1 consists of whole texts in six different languages: English, Italian, German, French, Polish and Russian. As we noted above, we only submitted results for English, Italian, German and Russian. We will elaborate on the reason for this in subsection 3.4. However, we still used the French and Polish datasets to augment our data, as we explain in the next section.

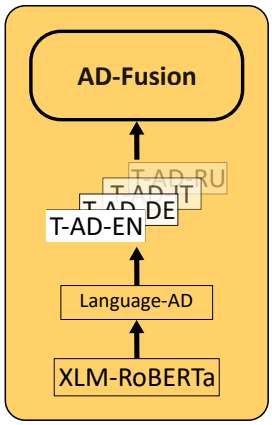
The main challenge of this subtask is presented by the limited availability, as well as the imbalance of the training data. Here, the opinion label is over-represented (76% of the labels) while the satire label is the clear minority class (5.8% of the labels). On top of that, the texts are often longer than the 512 token-limit BERT-type systems can handle. Consequently, utilizing all of the available data requires preprocessing the training data to address these issues.

Since the dataset for each language is fairly small but we have data for multiple languages, it makes sense to try and combine the distributed knowledge that can be extracted from the training data. To fully exploit the information contained in all languages, we use AdapterFusion (Pfeiffer et al., 2020a), which combines task-specific representations using a task-composition layer. We will elaborate more on adapters and AdapterFusion in the next section.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1166–1171



(a) Schema of Task Training and Transfer.



(b) Schema of approach for AdapterFusion

Figure 1: Approaches for zero-shot predictions with a pre-trained task-adapter and for combining task adapters for AdapterFusion. Note that these are schemata, the adapters are actually located within the XLM-R model, behind the feed-forward network (see (Pfeiffer et al., 2020a)) (a) Top: The task-adapter is trained using German data. Bottom: To do a zero-shot prediction for a different language (in this case, Spanish), the language-adapter is exchanged with the one from the zero-shot-language. (b) Several task adapters can be combined in parallel to train an AdapterFusion layer which learns to assign different weights to the task adapters depending on the input it is trying to predict.

3 System Overview

As mentioned, we use an adapter-based model to classify the news data. Adapter-based frameworks have shown to be very useful in dealing with multilingual problems, as their modular structure makes it possible to adapt task-specific representations to different languages (Pfeiffer et al., 2020c). Before we get into the details of our experimental setup, we want to elaborate on different measures we took in order to augment our data, as well as introduce the concepts of adapters and adapter fusion.

3.1 Data Augmentation

As a first step to augment our data, we decided to translate the training data from each language to all the languages we aim to build a model for. For this, we used the OPUS-MT models published in (Tiedemann and Thottingal, 2020) from the huggingface-hub[1]. We translated each sentence separately, using nltk's sentence-tokenizer to split the texts into sentences (Bird et al.). For some language pairs, no translation models were available. In these cases, we did not translate the data. For English, we translated the data from all other languages. For German, we were missing the Russian translation model, for Italian in addition to the Russian, we were missing the model for translation to Polish. Finally, for Russian, the only translation models available on the huggingface-hub are the ones to English and French.

Since the average text of the training data is longer than the 512 tokens BERT-type models can work with, we further augmented our data by dividing each text into multiple subtexts, depending on the length of the text. Each of these subtexts was then used as a new row with the same label as the original text in our training data.

3.2 Adapters

Adapters offer a modular and parameter-efficient solution for fine-tuning a base-model for specific tasks (Houlsby et al., 2019) or transfer task-specific knowledge to different languages (Pfeiffer et al., 2020c). Since subtask-1 consists of training data in multiple different languages, but the amount of data for a single language is rather small, we approached this problem with the plan to use adapters as a vehicle to combine the knowledge available in each language.

Adapter modules are added into the layers of pre-trained models and aim to learn one specific task without changing the weights of the pre-trained model (Pfeiffer et al., 2020b). Similarly to training adapters for tasks, it is possible to train language-specific adapters by adding an adapter to a multilingual base model and subsequently training the adapter using MLM (Pfeiffer et al., 2020c). When training a task-adapter with a multilingual base model, it is useful to also use a fixed language-adapter as it captures language-specific knowl-

[1] https://huggingface.co/Helsinki-NLP

edge, thus improving the final performance (Pfeiffer et al., 2020c). Transferring the knowledge from the trained task adapter to a different language is quite straight-forward: One simply needs to exchange the language-adapter with the language-adapter of the target language. We will use this setup for the zero-shot languages. A schema for this approach is displayed in Figure 1a.

3.3 Adapter Fusion

As explained in the previous subsection, adapters are modular and can be used in tandem with one another. But stacking them, as we are doing with language- and task-adapters, is only one display of this. A different way of combining adapters is called AdapterFusion. In AdapterFusion, different adapters are used in parallel in order to combine their task-specific knowledge (Pfeiffer et al., 2020a). AdapterFusion has been shown to be especially beneficial for tasks with small datasets, as it can rely more on the tasks that were trained on the larger datasets. However, our augmented data-sizes are fairly similar for most of the languages. Consequently, it is not clear if AdapterFusion will lead to performance gains or not.
Our approach for training AdapterFusion is to first train a task-adapter for the data in each language as explained in the previous subsection. Afterwards, we train AdapterFusion by stacking all the task-adapters in parallel on top of a language-adapter, as shown in Figure 1b.

3.4 Experimental Setup

To build the structure of our model, we used the AdapterHub introduced in (Pfeiffer et al., 2020b). This hub offers a wide range of pre-trained adapters, as well as a library to create new adapters and train them. Importantly, one can find many language-adapters in the AdapterHub. As our base-model, we use XLM-RoBERTa, which has shown good performances among other multilingual models (Conneau et al., 2019).
For each language, we then train a task-adapter by first stacking a language-adapter on top of the XLM-R model, before stacking a task-adapter on top of the language-adapter (this setup is explained with more details in (Pfeiffer et al., 2020c)). Because the AdapterHub does not have language-adapters with the XLM-R architecture for French or Polish, we only trained adapters for the other four languages.
As a baseline to the multilingual approach, we also trained several task-adapters which used monolingual models as their base. This baseline should yield similar results to simply fine-tuning the base-model itself, although it doesn't permit us to use a language-adapter for predictions on the zero-shot languages. We used the following base models: For English we used RoBERTa-base (Liu et al., 2019), for Russian we used ruBERT (Kuratov and Arkhipov, 2019) and for Italian[2] and German[3] we used models from the huggingface-hub.
We trained each task adapter using a learning rate between $5 \cdot 10^{-5}$ and 10^{-4}, with a batch size of 8. We used a weighted cross-entropy loss function to account for the imbalanced data.
After training a task-adapter for each language, we trained AdapterFusion by stacking all the task-adapters in parallel on top of a language adapter (see Figure 1b). We trained an AdapterFusion layer for each language, because to train a single AdapterFusion for all languages, one would have to change the model configuration by switching out the language-adapters during training, which is not straightforward. For the training of AdapterFusion, we applied a learning rate of $5 \cdot 10^{-5}$, a batch size of 8 and again used a weighted loss function.

4 Results

In Table 1 we show the results of the various model-configurations on the evaluation- and test-set and highlight the results we submitted to the task organizers with a dagger-symbol. In addition to the approaches with a multilingual base-model we show the results of the baseline which uses a monolingual base-model and fine-tunes a task-adapter stacked on top of it.
In order to be as clear as possible, we divide this section into two parts - the first describes results submitted before the task-deadline while the second describes results achieved after the deadline.

4.1 Pre-Deadline Results

The results of the models for the evaluation-set are mixed: For English and Russian, the configuration using AdapterFusion achieves the best performance by far. However, for Italian and German, the AdapterFusion approach is worse than the approach which uses simply the task-adapter. The

[2] https://huggingface.co/dbmdz/bert-base-italian-cased
[3] https://huggingface.co/bert-base-german-cased

Configuration	EN	IT	DE	RU	ES	GR	Average
Evaluation-Set							
XLM-ADTask	0.3615	**0.6101**	**0.7333**	0.4895			0.5486
XLM-ADFusion	**0.4769**	0.5595	0.6530	**0.5551**			**0.5611**
Ro/BERT/a-ADTask	0.4189	0.4221	0.6889	0.4762			0.5015
Test-Set							
XLM-ADTask	0.4436	**0.4552**†	0.6109†	0.3561	0.3267†	**0.7504**†	0.4905
XLM-ADFusion	**0.5942**†	0.4375	**0.6996**	**0.4257**†	**0.5795**	0.7502	**0.5811**
Ro/BERT/a-ADTask	0.5767	0.4112	0.6374	0.3738			0.4998
Final Rank	4	10	11	11	10	3	8

Table 1: F_1-macro scores of our different model configurations on the evaluation (top) and test-set (bottom). Scores marked by a dagger-symbol were submitted before the task-deadline and represent our scores on the leaderboard.

baseline approach generally achieved similar performances to the task-adapter, although it is much worse for the Italian language.

We always used the model that achieved the best performance on the evaluation-set, marked in bold in Table 1, to predict on the test-set. Comparing the results between evaluation- and test-set, the absolute value of the F_1-macro score changes a lot for all languages. For English, it is more than 10 percentage-points higher while it is significantly lower for all other languages, with up to 16 points for Italian. This suggests that the evaluation and test data are composed differently, however, since we do not have access to the test labels, we cannot investigate this further.

Overall, the German task-adapter achieved the highest F_1-macro score on the evaluation set, so we also used this adapter for the zero-shot languages. We applied the standard approach from (Pfeiffer et al., 2020c), replacing the German language-adapter with the zero-shot language-adapter and then predicting. The results of the zero-shot languages for the test-set were varied, with a high F_1-macro score for Greek, but a low value for Spanish. We didn't predict for Georgian, as, again, there is no language-adapter with the correct configuration (XLM-R) available for this language on the adapter-hub.

4.2 Post-Deadline Results

Since the F_1-macro-scores for the test-dataset were only released after the deadline, we did some more tests to see how different configurations performed. The AdapterFusion setup of the zero-shot languages displayed in Table 1 uses the German AdapterFusion as its base while only replacing the language-adapter. We can see that in the case of German, the configuration that performed the best on the evaluation set actually performs worse on the test set. The same can be observed when using the German task-adapter to predict for the zero-shot language Spanish, as the AdapterFusion approach shows a better result. Overall, the setup using AdapterFusion performs best on the test set in four out of six cases, although the performance in the Greek case is fairly similar for both configurations. In the last column of the table, we calculated the average score for all languages with values. We can see that for both the evaluation, as well as the test-set, AdapterFusion has the highest average score.

5 Performance Analysis

5.1 Performance in Zero-Shot scenario

We did a series of experiments to test the zero-shot predictions when transferring knowledge from different languages and show the results in Table 2. For each language, we transferred once using simply the task-adapter, and once using the AdapterFusion setup. In addition, we calculated the average values of all results from the task-adapters and the AdapterFusions and show them in the last column. For almost all results, using AdapterFusion improves the results in the zero-shot language. There are only two cases where this is not the case: For Greek, when transferring from German and when transferring from Italian, although it is very close in the first case.

In general, one would expect the performances of the different AdapterFusions to be similar, since they all consist of the same task-adapters. The only difference between them should be the language of the training data with which they were trained. In Table 2, we can see that the AdapterFusion re-

Zero-Shot Language	EN		DE		IT		RU		Average	
	Task	Fusion	Task	Fusion	Task	Fusion	Task	Fusion	Task	Fusion
ES	0.4066	**0.4325**	0.3267	**0.5795**	0.3353	**0.5045**	0.4318	**0.4832**	0.3751	**0.4999**
GR	0.3573	**0.6079**	**0.7504**	0.7502	**0.6807**	0.6521	0.3258	**0.6110**	0.5286	**0.6553**

Table 2: Comparison of the F_1-macro scores when transferring knowledge to the zero-shot languages (rows) from different origin-languages (columns).

sults for a zero-shot language are in a similar range: For Spanish, they vary between 0.4325 and 0.5795 and for Greek, the values lie between 0.6079 and 0.7502. However, considering they use the same task-adapters in their configuration, this is still a rather large difference. Because of this, we decided to take a closer look at the AdapterFusion attentions with regards to the different task adapters.

5.2 AdapterFusion Attentions

The AdapterFusion layer attends over the different task-adapters when an input is passed through it and activates them accordingly. By investigating these attentions, we can find out how exactly the Fusion layer creates its predictions depending on the input data.

First, we look at the AdapterFusion activations when predicting for the test-data. In Figure 2, we have depicted the mean attentions over the first eleven XLM-R layers for the test-data. We excluded the last layer, since (Pfeiffer et al., 2020a) found that its attentions show some unpredictable behavior because they are located right before the prediction head and not, like the other layers, between the frozen pretrained layers of XLM-R.

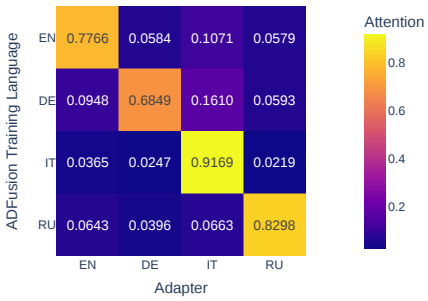


Figure 2: AdapterFusion attentions for the test-data.

We can see that generally, the AdapterFusion tends to activate the adapter of the language it is currently training with. However, especially for Italian, the fusion layer nearly doesn't activate any other task-adapters at all. It appears that the adapters of the other tasks are detrimental in this case - this is supported by the results in Table 1, where the task-adapter for Italian achieves better results than its AdapterFusion.

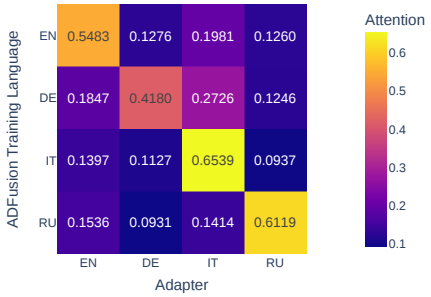


Figure 3: AdapterFusion attentions when transferring to greek.

In addition to the activations for the test-set, we take a closer look at the AdapterFusion attentions while predicting for one of the zero-shot languages, Greek, in Figure 3. We see a similar effect as in Figure 2, meaning the AdapterFusion attends most to the task-adapter of the language it was trained with, which explains the discrepancy between the AdapterFusion-results in Table 2. However, those attentions are generally lower than when predicting for its own language on the test-set. In contrast, the activations of the non-diagonal task-adapters are now much higher, indicating that the fusion now combines the different task-adapters more effectively.

6 Conclusion

We successfully investigate multiple approaches to determine the genre of news articles. We find that using a configuration which utilizes AdapterFusion to combine the knowledge learned from datasets in various languages delivers the best results for most languages. By comparing the attentions of the AdapterFusion between a zero-shot language and a trained language, we see that it effectively composes the learned knowledge when used to predict for an unseen language.

Further work could try to improve the setup of the

AdapterFusion, in order to make it possible to train it with multilingual data at once. In addition, including more languages (for example French and Polish, after training language-adapters for them), might improve the results, as more knowledge would be available.

References

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv*.

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. *arXiv*.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. *arXiv*.

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*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. *arXiv*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020b. AdapterHub: A Framework for Adapting Transformers. *arXiv*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020c. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. *arXiv*.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 Task 3: Detecting the Category, the Framing, and the Persuasion Techniques in Online News in a Multilingual Setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, SemEval 2023, Toronto, Canada.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – Building open translation services for the World. *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*.