DOI: 10.15439/2023F8078
ISSN 2300-5963 ACSIS, Vol. 37

Ausklasser - a classifier for German apprenticeship advertisements

Kai Krüger
German Federal Institute for Vocational Education and Training
Email: kai.krueger@bibb.de

***Abstract*—The German labor market system heavily relies on apprenticeships. Online Job Advertisements (OJAs) become an increasingly important data source to monitor labor market. Commonly, researchers use Information Extraction (IE) methods from Natural Language Processing (NLP) to extract entities such as skills and tasks from OJAs and draw conclusions about the labor market by aggregating them based on relevant variables such as occupations. Depending on the research question, it may be valuable to be able to exclude apprenticeships from these analyses, because apprentices will not be expected to have a specialized skill-set yet. As a result, Apprentice OJAs (AOJAs) may not reflect the dynamics in occupations and labor market as much as Regular OJAs (ROJAs). Furthermore, certain analyses may benefit from examining apprenticeships exclusively. This paper provides an efficient distilBERT based text classification model for this task and discusses findings from an experiment pipeline designed to identify the best possible implementation strategy of this task given the current NLP toolkit.**

I. Introduction

ONLINE Job Advertisements (OJAs) have been used to monitor labor market [3] with regard to dimensions such as skills and tasks [12], working tools [11], education [2], the impact of the covid pandemic [20, 13], specific industry sectors [16] and others. It is safe to say that OJAs are a valuable data source for monitoring labour market for years to come. Methodologically, researchers usually use Natural Language Processing (NLP) and Information Extraction (IE) to gain insights into the contents of the OJAs to aggregate information about entities such as skills or tasks based on dimensions they are interested in such as occupations or industry sectors.

Being able to exclude Apprenticeship OJAs (AOJAs) or consider them exclusively is an important variable for these analyses, especially in Germany. The goal of this paper is therefore to develop a text classification model, that can predict whether an input OJA is an AOJA or a Regular OJA (ROJA), and make it publicly available[1]. It also contributes by conducting experiments to find best way to implement such a model within the current NLP landscape. Other researchers can use the findings to build their own models for other languages or similar tasks. Specifically, it tests for eight parameters that concern composing the training data, model choice, hyperparameter choice and task modeling. For these parameters, hypotheses are formulated and tested in a random search with 100 trials. The final parameter setup is then reported for another 13 runs to reduce the effect of randomness and the model is published publicly. The structure of the remaining paper is as follows. In Chapter II we briefly explain the need for an AOJA classifier. In Chapter III we frame building this classifier as a NLP problem and introduce the different choices we had to make when constructing such a classifier. Corresponding to these choices we formulate hypotheses. The choices can be represented by parameters in our experiment pipeline, which is explained in Chapter IV. Chapter V presents the results and Chapter VI discusses them and mentions major limitations of the experiment. Finally, Chapter VII concludes this paper and gives suggestions for further research.

II. Building an AOJA classifier for labor market research

The apprenticeship system is a key factor for the German labor market. Although higher education gets increasingly important, vocational education and training (VET) makes up about half of the German post-secondary education system [4, 5]. The VET system provides the labour market with skilled workers and is an established process for the transition from school to work. Since VET is still part of people's education, apprentices are not expected to have a major skill-set yet that companies could demand in their ads. Furthermore, due to the formalization of VET in Germany, tasks listed in AOJAs are likely to be more generic. Therefore, AOJAs do not reflect labour market dynamics as much as ROJAs. Inversely, AOJAs isolated are a valuable source to ask research questions specifically with regard to apprenticeships. Research questions could include finding out how employers try to attract apprentices or predicting trends in the popularity of certain occupations based on AOJA number development. To the best of our knowledge, across languages there is no model currently available to classify OJAs as to whether an apprentice is being sought or not. Papers analysing labor market on the basis of OJAs have so far not made any distinction between AOJAs and ROJAs. Few publications use OJAs to specifically conduct research about apprenticeships and those that do use a smaller size of hand selected ads [9, 8]. The primary goal of this paper is therefore to build and publish an AOJA classifier. In the next chapter we frame this task as a binary text classification problem and discuss a variety of aspects that ought to be considered in the process of constructing the model given the current NLP landscape.

[1] Model: https://huggingface.co/KKrueger/ausklasser
Code: https://github.com/KruegerETRF/ausklasser

III. AOJA classifier as a NLP problem

Text classification is a well explored NLP problem. With the transformer architecture and Hugging Face infrastructure powerful off the shelve solutions are available with minimal time invest and coding efforts. At the same time, the particular task of classifying AOJAs and ROJAs has not been explored and no datasets or benchmarks are available. In an explanatory data analysis [19] it was found that the texts show distinguishable characteristics for the task at hand. In most cases an AOJA explicitly states that an apprentice is being sought. This not only makes it plausible to build an AOJA classifier, but also makes it a comparably easy task. So, then the question is, whether it is sufficient to simply take any pretrained model on Hugging Face and finetune it on some hundred samples and be done?

We argue that there is still a variety of decisions to be made to tackle this problem in the most optimal way given the current state of NLP. We include different dimensions into the evaluation of our model including robustness, epistemological validity, efficiency, ethical concerns, flexibility and generalizability. We derive research questions from these decisions that we formulate as hypotheses that are being empirically tested via the experiment pipeline described in section IV. In that sense, this paper serves as a reference point to other researchers facing similar problems. The structure of this chapter is to formulate the hypotheses and then discuss their background and relevance.

Hypothesis H_1: There will be no difference between multi- and monolingual pretrained models.

Hypothesis H_2: Domain adapted models will perform better than generic models.

Hypothesis H_3: Lighter models will perform equally well to bigger models.

The first three hypotheses deal with the choice of the pretrained model, which is the first decision to be made. The most obvious goal is to choose the best performing model for the given task. A first reference point could be the standard BERT model [7], which has been trained multilingually[2], including German texts. Then, there are monolingual models for the German language like [6]. Now, given the task at hand, which one performs better? Hypothesis H_2 is concerned with a BERT model domain adapted to German OJAs developed by [10]. It is plausible to expect that this model performs better, because it has already internalized patterns of OJA and might be able to generalize quicker, for example by knowing relevant synonyms for the German word for apprentice, *Auszubildender*, (e.g. *Azubi*) or other keywords.

Both of these research questions have the primary goal to find a very robust and well performing model in model construction. Beyond this, however, there is another relevant aspect for model choice: computational cost. Higher computational cost means that model training is more expensive financially and has an increased environmental impact [18]. Even though in [18] models are being trained from scratch, fine-tuning models is also costly. Additionally, the efficiency of the trained model at inference is a relevant factor for research institutes and companies with a smaller budget. Generally, the more efficient a model is the better as long as its performance does not suffer. Since the task at hand is rather simple, it is plausible that lighter models such as [17] perform equally well, which is tested by Hypothesis H_3.

Hypothesis H_4: Hyperparameter search can be neglected

Also intertwined with the points about training cost is the search for optimal hyperparameters. The more different setups are tried, the more resources need to be used. Therefore, the authors in [18] suggest to use hyperparameter optimization algorithms. The study in [14], however, shows that these techniques can fail given an insufficient time budget and are prone to overfitting. Given the simplicity of the task, the question is, if it is even necessary to perform extensive hyperparameter search or if using default or common configurations is sufficient. Hypothesis H_4 therefore tests whether a single model hyperparameter consistently affects model's performance. As we will see in section IV the pipeline chooses hyperparameters so that the search space only affects commonly used values (including default values). Given the simplicity of the task the hypothesis is that it does not matter significantly, which learning rate, for example, is chosen. Certainly, choosing absurd values for the learning rate would affect models performance significantly, but this is not relevant to the hypothesis. With the regard to learning rate it has to be mentioned that all models (and configurations) explored in this experiment pipeline use the adaptive gradient algrotihm [15] AdamW, which means the importance of the initial learning rate hyperparameter decreases over time.

Hypothesis H_5: Given two datasets D1 and D2 from different sources and substantial textual differences, models trained primarily on D1 data will perform better when testing on D1 data and vice versa.

Hypothesis H_6: Using more than two labels will increase the robustness of the model on downstream binary predictions

The datasets used are described in Appendix A in detail. For Hypothesis H_5 it is important to know that there are two different labeled datasets available that each are different structurally. Specifically, one of the datasets (D1) has a lot of boilerplate remains from the scraping process, whereas the other one (D2) has only cleaned text without boilerplate, containing only the actual ad. Cleaning D1 is not an option. Also, D1 comes from various online sources, whereas D2 comes from the same source website, which might lead to other biases/differences, such as D2 being more homogeneous linguistically or in terms of labor market specific factors (ie. certain industry sectors are more likely to appear in D2 than

[2] Of course, there is also the monolingual English version, but that is irrelevant in this case.

others compared to D1). The goal for our model is to not be influenced by such factors. Ideally, it generalizes over any German OJAs fed into it. Preliminary analysis showed, however, that simpler models trained on D2 perform worse on D1 data. Therefore, Hypothesis H_5 tests if such effects are also true for transformer based models. This aspect is also relevant in the context of publishing our model publicly. Researchers might be able to trust its performance on their data more, if we can falsify Hypothesis H_5.

Another aspect of the two datasets is that they are labeled more fine grained than only AOJA and ROJA. There are further categories like internship or leading position. These categories, however, are not common between both datasets. Also, the most important goal of the classifier is to distinguish between AOJAs and ROJAs. It would, however, be a potential future advantage, if some of the other classes could also be identified by our classifier. With regard to Hypothesis H_6 another factor has to be considered: given the high amount of OJAs that seek "regular" workers in both datasets, binary set ups will often end up without many other "special" categories such as internships. This might lead to the model not learning to differentiate between AOJA and non-AOJA, but between ROJA and non-ROJA positions. This would be prevented by the more sophisticated categories. Note, that in case a multi-class model was trained its predictions were still aggregated to do binary classification during test phase (see section IV for further details).

Hypothesis H_7: Balanced datasets will perform better than unbalanced datasets

Hypothesis H_8: Models will perform well with limited training data

The amount of AOJAs compared to ROJAs is much smaller (rougly 14 percent). There is no sophisticated balancing in place such as data augmentation methods, but we will compare simple over- and undersampling to not balancing data and see, how this influences the performance on a balanced testset.

Another question when building a model for a new NLP dataset where no benchmarks exist yet is how much data labeled data is required. Since we have a lot of labeled data available, we can test different sizes up to 10.000 ads, but we hypothesize that given the simplicity of the task models should be able to achieve good results with limited training data. See section IV for further details on how much data is used exactly.

IV. Experiment Pipeline

In this section we describe the experiment pipeline. Based on the hypothesis described in section III there are eight parameters introduced to the pipeline, three of which are common hyperparameters fed into model training. Table 1 shows the parameters and their search space. The pipeline consists of three steps explained below. The parameters are inserted in the first two steps (compose data and training), whereas the last step (testing) reports the final metrics. It is important to mention that these metrics are always tested against the same testset, regardless of how the training (and evaluation) data ended up after the first step. Fig. 1 gives an overview of the pipeline. In the initial experiment a random search with 100 trials was performed.

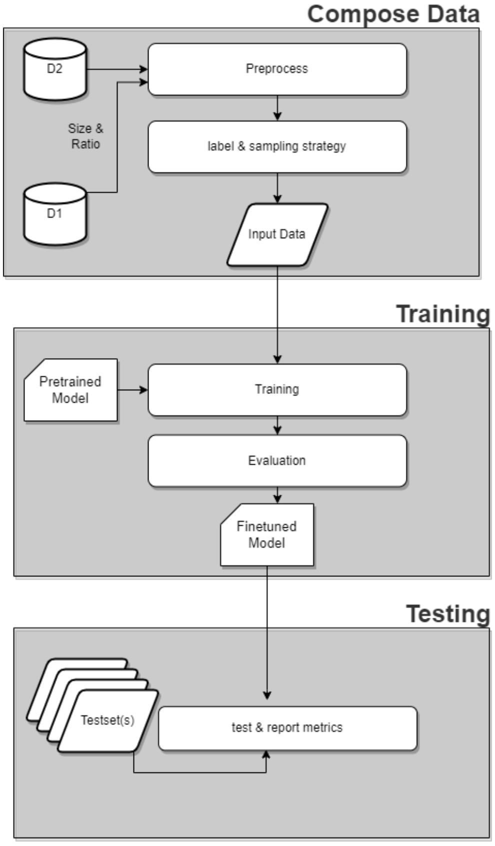

Fig. 1. Simple visualization of the experiment pipeline

A. Step 1: Compose Data

The compose data step consists of accessing the data from the two datasets with regard to the size and ratio parameters. It then harmonizes both datasets into one and performs a check to prevent any ads that are used in the testset later to be included in the training data. Then it aggregates labels based on the label strategy chosen. For the binary option all non-AOJAs are being aggregated into one category (ROJA). For the multiclass option two additional label classes are added. They are described in more detail in the appendix. Finally, it considers the sampling strategy, either leaving the data as is (no balance) or performing over- or downsampling. Both work similar. The highest/lowest number of instances for a class is located and then data is either randomly duplicated (oversampling) or removed (downsampling) for all other classes until that number is reached. The final dataset is then being forwarded

TABLE I Parameters and search space

Parameter	Options
model	multilingualBERT, germanBERT, jobBERT, distilBERT
size	100, 500, 1000, 5000, 10000
ratio	1, 0.7, 0.5, 0.3, 0
label strategy	binary, multiclass
balance strategy	oversample, downsample, no balance
learning rate	0.0001, 0.00001, 0.000001
epochs	3, 5, 7
warmup	0, 500

to the training step. [3]

B. Step 2: Training

The training step loads in the pretrained tokenizers and models from the options listed in Table 1 as well as the input data from step 1. Then, the standard fine-tune process is being initiated, where the above described hyperparameters are being varied. A statement about hardware used and the energy consumption can be found in the appendix. Most notable is that the batch size had to be kept relatively small (eight) due to hardware limitations, which potentially hinders performance.

The dataset is split 0.7/0.3 for evaluation during training. For evaluation accuracy, precision, recall and f1 are being measured and logged along with the training loss. When there are four labels precision, recall and f1 are being averaged via the macro average method. Once the training is done, the model with the fine-tuned weights is saved and forwarded to the testing step.

C. Step 3: Testing

In this step, the saved model from the training steps is loaded and tested against the independent testset as described above. The testset contains 80 job ads split 40/40 between D1 and D2, and then split 20/20 between both classes. To ensure validity of the testset, all ads have been reevaluated blindly by two annotators each. In case the model was trained on multiple classes its predictions were aggregated to the binary labels.

Like in training the metrics accuracy, precision, recall and f1 were being used. Since the testset is balanced, accuracy can well indicate model performance. For the other metrics AOJAs have been labeled the positive category, because it is more important. Each metric was measured four times:

- For the entire testset
- For testset data only from D1
- For testset data only from D2
- For testset data whose texts surpass the 512 max token length that the models pose

Building sub-datasets to include only data from D1 or D2 respectively was to study the effect on input data specifics on the model's ability to generalize and test Hypothesis H_5. To make the model more robust, a dataset with only those texts that contained more than 512 tokens has been build to ensure that the models performance is not influenced by truncation.

V. Results

The above described experiment pipeline has been used multiple times with different purposes to test the hypothesis or to build the final model. This chapter is split into two subsections. First, the initial search with 100 runs to test the influence of the different parameters is described and reported. Then, the configuration of the final model is described and the results of 13 runs are reported.

A. Initial search

The initial search was a random search with 100 experiment runs randomly selected from the 10.800 possible parameter combinations possible. The goal was to get an overview over the general performance and different parameters. Of these 100 experiment runs, 8 ended unfinished due to hardware issues. The exact parameter configurations and corresponding results can be accessed in the repository. Fig. 2 shows distribution of experiments for different metrics. We can observe a very strong fluctuation in experiment results, ranging from 0 F1 in the testset to 1.0. Further analysis shows that models with 0 F1 tend to have 0.5 accuracy, which leads to the conclusion that the model has overfitted into always predicting one category. Generally, the three parameters size, label strategy and balance strategy can lead to datasets where there either not much data left at all (because of downsampling) or data is heavily imbalanced so that the model performs well in training just by predicting the largest category (normal OJAs). However, this is not true for all cases. For example, run *66c90a54* [4] had 1.250 samples for both categories (downsampled from 10.000 overall), which should be more than sufficient to learn a meaningful representation of the two classes. Also, accessing training metrics showed that these models do have reasonable performance in the evaluation. Further analysis, however, showed that exploding gradients were likely a problem in these cases. Based on this we can already falsify hypothesis H_4. Choosing the correct hyperparameters based on the setup **is** important as it can prevent overfitting. Especially the number

[3] Note, that for this step the code is only partially published, because the data is not published and the access to the database works with internal procedures. In the published repository this part is replaced with pseudo code

[4] individual runs can accessed from the repository

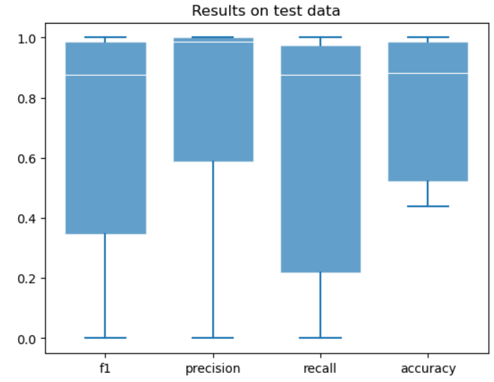

Fig. 2. Results of the initial search on testset performance by different metrics. The white line indicates the median.

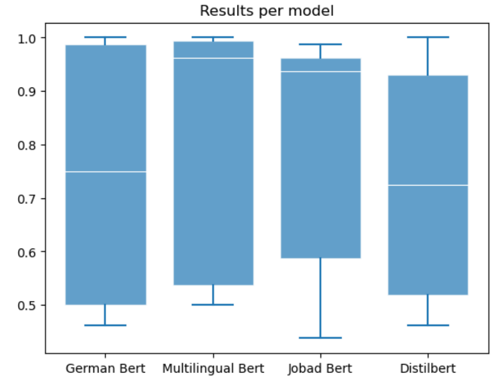

Fig. 3. Accuracy of all runs per model on testset performance. The white line indicates the median.

of epochs mattered significantly. Overall, the amount of models not working downstream while showing reasonable metrics during training was very high, which proves the importance of using a separate testset. Due to the high amount of outliers, further analysis will prefer to analyse median over mean values.

Overall no single parameter consistently performs bad. Each single parameter achieves accuracy scores > 0.9. The median for accuracy is 0.88, which confirms that good performing models can be build in a variety of ways. Also, several runs achieved accuracy on the testset of 1. Generally, fig. 2 also shows that precision is higher than recall most of the times. This might again be due to imbalanced datasets where the model learns to prefer predicting ROJAs. Another general observation is that overall models did not perform worse on longer truncated texts.

The first three hypotheses all deal with the choice of the pretrained model. The results show that the multilingual BERT model outperforms the monolingual model. The domain adapted model outperforms the monolingual model it was adopted from, but performs slightly worse than the multilingual model overall while, however, showing less deviation. The lighter distilBERT model also performs worse, on par with the German BERT model. In that sense, none of the first three hypotheses can be verified. With regard to hypothesis H_5 the results falsify the hypothesis. Fig. 4 shows more data from one dataset does not necessarily lead to better performance on the same data downstream. Specifically, the D1:D2 70:30 ratio performs better overall on D2 data in the test phase. However, also the balanced data performs better on D2 data. This might likely be due to D1 data containing boilerplate that confuses any model. With regard to the actual performance, 0 and 0.7 ratios perform best, which indicates that differences must also be attributed to other parameters. Therefore, a balanced 50:50 split still seems like the most reasonable option. hypothesis H_6 was verified based on these runs: multiclass models usually performed better, achieving 0.91 median accuracy, whereas binary models only got to 0.81 accuracy. Potentially this is because the additional classes helped prevent the models from overfitting.

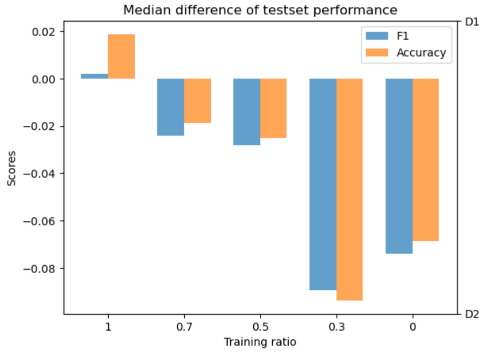

Fig. 4. Median difference of all runs per ratio on testset performance on D1 data versus D2 data. Positive values mean models performed better on D1 data, negative values mean models performed better on D2 data

In terms of balancing the data (hypothesis H_7) oversampling (0.94 median accuracy) and not balancing the data (0.93 median accuracy) performs much better than downsampling (0.58 median accuracy). The low performance of downsampling is however explained by the low amount of total data left when small datasets are being accessed to begin with. If considering only the experiments with at least 1.000 ads, for example, downsampling achieves 0.9 median accuracy. Considering only larger training sets for oversampling, however, also increases median performance to 0.97 accuracy. In any case we cannot confirm hypothesis H_7, because not balancing the data did not hurt performance significantly. With regard to hypothesis H_8 it shows that actually increasing data to several thousand samples still significantly increases performance and is the only major influence parameter. Although in some cases 100 or 500 sample setups showed good results, the overall performance increases with data size. This means that even for seemingly simple tasks researchers can likely increase performance by gathering more data.

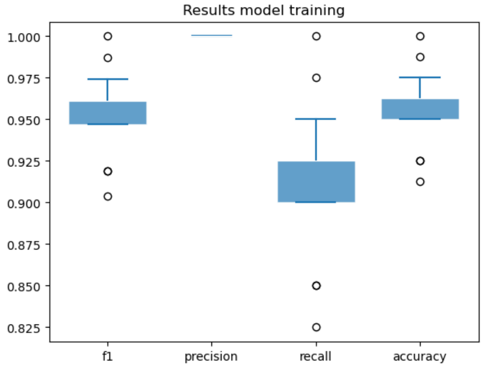

Fig. 5. Results of 13 experiments with fixed parameters

B. Training the model

Given the results from the initial search, how to construct the final model? Despite worse performance, it was decided to first try to use a pretrained distilBERT model, because given enough data those would still perform very well and have the benefit of having reduced cost. Since multilabels worked better initially, it was chosen here as well. For balancing, no balancing was decided upon, because it worked almost as well as oversampling and due to the lower amount of training samples is more energy and time efficient to train. The amount of data chose was 10.000, because a greater sample improved results in the initial search and the ratio was 0.5, which proved to be a robust choice. A learning rate of 0.0001 and no warmup steps were also chosen. For the epochs, a new value of 4 was introduced, because the analysis of training curve showed that models often needed more than three epochs, but sometimes started to overfit at five already. With these fixed parameters 13 runs have been performed to reduce the influence of randomness when reporting the final model. The results (Fig. 5) show a consistently good performance, but outliers still having a variance of roughly nine percent. Precision was higher than recall again, indicating a slight bias towards the ROJA category. Of the experiments a robust model was chosen and uploaded publicly. It achieved .98 accuracy on the testset and .9 accuracy in training evaluation (on four classes). All metrics can be accessed

VI. Discussion and Limitations

The primary goal of this paper was to publish a well performing model to classify German AOJAs and ROJAs. The model published achieved high results in training and testing and is able to classify German OJAs into four categories and AOJAs are one of them. Using the distilBERT architecture, our model is also small and efficient.

In terms of gaining insight into the decisions to be made when building such a model, there were a number of interesting findings. First, while the overall performance was good for many models, there were heavy outliers, which shows that experiments need to be conducted thoroughly and final training metrics cannot be taken for granted. Also, researchers need carefully select hyperparameters like the number of training epochs. Learning rate and warmup steps seemed to play less of an important role. Another key finding was that using more categories, if available, might boost performance for downstream tasks with less categories, because the model learns differntiations more explicitly. Also, the model was able to perform better on clean data, even if boilerplate data was favoured in training (70:30 split). If models trained on mostly clean text on the other hand get exposed to data containing boilerplate the performance drops significantly. Longer texts that had to be truncated did not pose an extra challenge for any of the models. This is not unexpected, because the information required to distinguish the texts for the task at hand is usually found in the opening section of the ads. Our findings suggest that unbalanced training data can also lead to good generalization, if there is enough data overall.

In terms of the models, every pre-trained model was able to produce good fine-tuned models. Some, however, did so more consistently than others. Given the factors described in Chapter III, it might be advisable to use smaller and more efficient models, despite them perform worse in some setups. As shown in Chapter IV B, they perform just as well with the correct setup and offer additional advantages. The most important parameter for any setup as to our findings was the size of the dataset. It showed that even for comparably simple tasks performance can be increased by increasing the data size.

The major limitation of the experiments in the initial search was that the hypotheses were not tested individually. When a certain parameter setting performed worse than another, for example a pretrained model performed worse overall than another, this might not have been the effect of that setting, but because randomly it had less favourable settings in other parameters such as size. Regarding size it also has to be mentioned that for low data settings no methods designed specifically for such cases [1] have been tested. Furthermore, the size of the testset was relatively low given the amount of labeled data that was used during training. This is because the quality of the labels was questionable and testset labels have been reevaluated manually, see Appendix A. A larger testset would have been beneficial, but was not feasible given the resources available.

VII. Conclusion and Outlook

Despite the limitations mentioned above, the published model can be regarded as robust and usable for researchers analyzing German labor market with OJAs. Other researchers can council the findings of the experiment pipeline to make more profound decisions in model construction. Some of the findings here are also worth investigating further. Especially the effect of different datasets on the ability of models to generalize and the effect boilerplate generally has on NLP models is worth looking into. It is also important to keep this aspect in mind when using public models directly. How does the data from that model differ from the data it is supposed to be used on? Scraping boilerplate might be a significant pitfall here. Another aspect worth looking into is the strategy of using more labels than required for the downstream task at hand. Because the categories are more fine-grained, the models will be more robust downstream. The obvious downside of this approach of course is that it requires more data to begin with. But if a lot of data or resources to label additional data are available, adding additional categories might be a fruitful strategy to increase performance. Further investigation into these topics might include setting up more controlled experiments testing single parameters only across different tasks, datasets, models, etc.

Appendix A
Data

For this task, two labeled datasets are available to the authors. Both datasets are protected and cannot be published. The description here serves for transparency and reproducability. The two datasets are:

- Scraped-Data (D1): This dataset comes from a commercial provider of scraped OJAs from 2015 to 2022. In a project, roughly 15.000 OJAs have been labeled according to the type of worker being sought (including apprentices). The annotation process, however, was only a single blind annotation without further quality control. Also, this dataset suffers from unclean full texts, meaning that boiler plate and texts fields are often not separated from the actual ad and stored together as the full text in the data base.
- BA-Data (D2): This dataset is being provided by the Bundesagentur für Arbeit (BA). It consists of roughly 10 Million OJAs from 2011 to 2022 and comes with labeled metadata. Usually this metadata is of good quality, because it is hand labeled by the expert employees of the BA. However, it was not originally intended for scientific use and there are no further control mechanisms for label quality in place. Also, the metadata is not always consistent and label schemes may change over time. Another relevant information is that the full texts come manually cleaned and are free from any boiler plates or text fields that are often found in scraped data.

As shown, neither dataset is of undisputed validity. Therefore, it was decided to construct a test dataset of 80 OJAs that are equally split between the two datasets and among these splits are also equally split between the two categories. In other words there are 20 OJAs for each category for each dataset. This dataset was cross validated by two independent annotators (One male, one female, both German, one student in economics, one researcher in NLP). In some cases OJAs were too short to contain sufficient information or consisted of only boilerplate. These cases were removed. In all remaining cases both annotators agreed on the labels. This testset is being kept entirely out of model training, but each experiment run tests against it in the end.

Appendix B
Categories

Both datasets do not actually come in a binary labeled form, but have further categories of different job positions like internship or leading role. These categories differ between datasets and it is not possible to establish an unambiguous mapping. Of course, apprenticeships are a category in both datasets and the other labels can be aggregated to the ROJA category. However it was decided to include a set up with four different categories into the experiment pipeline. The categories then are as follows:

1) Apprenticeships
2) Other minor positions
3) Leading position
4) Regular workers

To see the exact mapping of other minor positions, please access the mapping dictionary in the utils.py file in the repository.

Appendix C
Hardware & Parameters

All experiments were run on a NVIDIA GeForece RTX 3080. Training time varied between roughly five and twenty minutes per run depending on dataset size and number of epochs. The batch size for training and evaluation were eight. All other hyperparameteres that potentially influence the models performance (and are not includeded in the experiment pipeline) were the defaults of the huggingface training arguments from the trainer class.

References

[1] Iz Beltagy et al. “Zero- and Few-Shot NLP with Pretrained Language Models”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 32–37. DOI: 10.18653/v1/2022.acl-tutorials.6. URL: https://aclanthology.org/2022.acl-tutorials.6.

[2] Phillip Brown and Manuel Souto-Otero. “The end of the credential society? An analysis of the relationship between education and the labour market using big data”. In: *Journal of Education Policy* 35.1 (2020), pp. 95–118. ISSN: 0268-0939. DOI: 10.1080/02680939.2018.1549752.

[3] Marlis Buchmann et al. “Swiss Job Market Monitor: A Rich Source of Demand-Side Micro Data of the Labour Market”. In: *European Sociological Review* (2022). ISSN: 0266-7215. DOI: 10.1093/esr/jcac002.

[4] Statistsiches Bundesamt. *Berufsbildungsstatistik*. Accessed on August 17, 2023. URL: https://www-genesis.destatis.de/genesis/online?operation=previous&levelindex=2&step=2&titel=Ergebnis&levelid=1690804374122&acceptscookies=false#abreadcrumb.

[5] Statistsiches Bundesamt. *Statistik der Studenten*. Accessed on August 17, 2023. URL: https://www-genesis.destatis.de/genesis/online?sequenz=tabelleErgebnis&selectionname=21311-0010#abreadcrumb.

[6] Branden Chan, Stefan Schweter, and Timo Möller. “German’s Next Language Model”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 6788–6796. DOI: 10.18653/v1/2020.coling-main.598. URL: https://aclanthology.org/2020.coling-main.598.

[7] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[8] Khristin Fabian and Ella Taylor-Smith. *How are we positioning apprenticeships? A critical analysis of job adverts for degree apprentices*. English. United Kingdom: Society for Research in Higher Education, 2021.

[9] Khristin Fabian et al. “Signalling new opportunities? An analysis of UK job adverts for degree apprenticeships”. In: *Higher Education, Skills and Work-Based Learning* ahead-of-print.ahead-of-print (2023). ISSN: 2042-3896. DOI: 10.1108/HESWBL-02-2022-0037.

[10] Ann-Sophie Gnehm, Eva Bühlmann, and Simon Clematide. “Evaluation of Transfer Learning and Domain Adaptation for Analyzing German-Speaking Job Advertisements”. In: *Proceedings of the 13th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, 2022.

[11] Betül Güntürk-Kuhl, Philipp Martin, and Anna Cristin Lewalder. *Die Taxonomie der Arbeitsmittel des BIBB: Revision 2018*. 2018.

[12] Robert Helmrich et al. *Berufsbildung 4.0 – Fachkräftequalifikationen und Kompetenzen für die digitalisierte Arbeit von morgen: Säule 3: Monitoring- und Projektionssystem zu Qualifizierungsnotwendigkeiten für die Berufsbildung 4.0*. 1. Auflage. Vol. 214. Wissenschaftliche Diskussionspapiere. Leverkusen: Verlag Barbara Budrich, 2020. ISBN: 9783962082024. URL: https://www.bibb.de/dienst/veroeffentlichungen/de/publication/show/16688.

[13] Jakob de Lazzer and Martina Rengers. “Auswirkungen der Coronakrise auf den Arbeitsmarkt: Experimentelle Statistiken aus Daten von Online-Jobportalen”. In: (2021).

[14] Xueqing Liu and Chi Wang. *An Empirical Study on Hyperparameter Optimization for Fine-Tuning Pre-trained Language Models*. 2021. arXiv: 2106.09204 [cs.CL].

[15] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

[16] Mirjana Pejic-Bach et al. “Text mining of industry 4.0 job advertisements”. In: *International Journal of Information Management* 50 (2020), pp. 416–431. ISSN: 02684012. DOI: 10.1016/j.ijinfomgt.2019.07.014.

[17] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: 1910.01108 [cs.CL].

[18] Emma Strubell, Ananya Ganesh, and Andrew McCallum. *Energy and Policy Considerations for Deep Learning in NLP*. 2019. arXiv: 1906.02243 [cs.CL].

[19] Dennis Ulmer et al. “Experimental Standards for Deep Learning in Natural Language Processing Research”. In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 2673–2692. URL: https://aclanthology.org/2022.findings-emnlp.196.

[20] Stefan Winnige and Alexandra Mergener. “Homeoffice-Boom im Zuge der Corona-Pandemie: Welche Potenziale zeichnen sich langfristig für akademisch und beruflich Qualifizierte ab?” In: *Berufsbildung in Wissenschaft und Praxis* 50.2 (2021), pp. 27–31.