Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 3892–3901
Marseille, 20-25 June 2022

Evaluation of Transfer Learning and Domain Adaptation for Analyzing German-Speaking Job Advertisements

Ann-Sophie Gnehm, Eva Bühlmann, Simon Clematide
Department of Sociology, Department of Computational Linguistics
University of Zurich
gnehm@soziologie.uzh.ch, buehlmann@soziologie.uzh.ch, simon.clematide@cl.uzh.ch

Abstract

This paper presents text mining approaches on German-speaking job advertisements to enable social science research on the development of the labour market over the last 30 years. In order to build text mining applications providing information about profession and main task of a job, as well as experience and ICT skills needed, we experiment with transfer learning and domain adaptation. Our main contribution consists in building language models which are adapted to the domain of job advertisements, and their assessment on a broad range of machine learning problems. Our findings show the large value of domain adaptation in several respects. First, it boosts the performance of fine-tuned task-specific models consistently over all evaluation experiments. Second, it helps to mitigate rapid data shift over time in our special domain, and enhances the ability to learn from small updates with new, labeled task data. Third, domain-adaptation of language models is efficient: With continued in-domain pre-training we are able to outperform general-domain language models pre-trained on ten times more data. We share our domain-adapted language models and data with the research community.

Keywords: Text Mining, Transfer Learning, Domain adaptation, BERT, Computational Social Science, Job Advertisements

1. Introduction

Large-scale automated text analysis is becoming more and more standard in digital scholarship. For common text mining tasks, for instance, Named Entity Recognition and Linking on newspaper texts or Sentiment Analysis of social media content, several off-the-shelf solutions exist either as research software or commercial services. However, for many specific research questions, application domains and languages, specialized natural language processing (NLP) models still need to be trained.

Effective training of high-quality text mining models without much annotation effort is now possible thanks to *transfer learning*, which is typically divided in a pre-training and fine-tuning phase. Over the last years, this machine learning technique, which makes clever use of large amounts of raw text for pre-training, evolved successfully: starting with static lexical word embeddings as word2vec (Mikolov et al., 2013) followed first by contextualized embeddings encoded by recurrent neural network-based conditional generative language models as ELMo (Peters et al., 2018), and then shortly after introducing BERT embeddings (Devlin et al., 2019) where representation learning relies on a transformer-based denoising language modelling pre-training task. Although minor architectural changes have been applied to the original BERT idea, this approach is still state of the art in NLP.

The expected benefits of transfer learning approaches are twofold: First, given the pre-trained text representations, fine-tuning needs much less task-specific training material to achieve high performance. Second, fine-tuned models are more robust and less brittle, thus often avoiding hard errors or even systematic errors, which were typical for pattern-based text mining (Hedderich et al., 2021; Ruder, 2019). This is especially useful in domains with a highly dynamical language development.

This paper presents text mining approaches on German-speaking job advertisements from Switzerland. Not surprisingly, jobs, job ads and their language are changing rapidly in a world of globalization and digital transformation, thus leading to a noticeable data shift. A printed job advertisements in a newspaper from 1990 and an online job ad from 2020 differ in text length and other characteristics of their publication channel. The work presented here is part of a computational social science project that seeks to answer sociological and economical questions about the development of the labour market over the last 30 years by analysing large collections of job ads.

The texts in job ads are not only rapidly changing, they are also very particular in their structure and formulation. Therefore, another important aspect of our work is the adaptation of models that were originally pre-trained on general purpose or mixed domains via continued pre-training and vocabulary customization. How much improvement with which domain adaptation techniques is feasible for typical machine learning tasks in the context of transfer learning? Can more task-specific training data compensate for a lack of domain adaptation during fine-tuning?

The relevant information needs of applied sociological and economic research on the demand in the Swiss labour market that we consider in this work can be summarized as follows: a) What profession is this ad looking for? b) What is the main task for the jobseeker? c) Is work experience required? d) What skills are required, in particular, what information and communications technology (ICT) skills?

We operationalize the questions (a-c) directly as document classification tasks on job advertisements and report their experimental results. Question (d) is more difficult, and we divide it into two more general text analytics processes: First, a token-based sequence labelling task that segments job ads into different content zones (e.g. company description, hard or soft skills). Second, a term extraction task that identifies single- and multi-word ICT expressions.

Our various experiments with different types of machine learning problems give a broad assessment of modern transfer learning and the expected benefit of current domain adaptation techniques. Additionally, we make our datasets and models available for further experimentation in the research community.[1]

The remaining of this paper is structured as follows: Section 2 discusses related work. Section 3 describes our datasets. Experiments and results are presented in Section 4. Section 5 summarizes our findings and provides directions for future work.

2. Related Work

The adaptation of general-purpose language representations models (LM) like BERT (Devlin et al., 2019) to specialised domains has been an active area of research in the last years. Starting with work on scientific texts (Beltagy et al., 2019), the biomedical (Lee et al., 2019) and the legal domain (Chalkidis et al., 2020), lately there is a body of research experimenting extensively on pre-training, continued pre-training and fine-tuning language models for special domains (Alrowili and Shanker, 2021; Gu et al., 2021; Lewis et al., 2020; Gururangan et al., 2020). We provide a brief overview of the different approaches.

Two approaches in building domain-specific LMs can be distinguished: First in *continued pre-training*, we start with a general purpose LM like BERT, typically pre-trained on a large collection with a mixture of text types. This means that the (sub)word vocabulary of this base model – for instance computed statistically by the SentencePiece (Kudo and Richardson, 2018) or BPE (Sennrich et al., 2016) algorithm – is used, and model weights are updated during continued training on domain-specific corpora. Some general purpose LMs reserve a certain amount of vocabulary slots for further pre-training, which then can be filled with domain-specific subwords. Second in *domain-specific pre-training*, the model is built from scratch with text material from the application domain, typically on a collection of smaller size compared to general LMs. This means that the LM is well-informed about the most frequent words or subword units of the domain. While this latter approach has the advantage of providing a domain-specific vocabulary, it needs much bigger amounts of in-domain data, since it cannot benefit from any preceding general language model training.

Recent work has evaluated different approaches in building domain-specific language models on a range of downstream NLP task. Experiments assess different transformer architectures, further pre-training versus training from scratch, and include often grid searches for fine-tuning hyper-parameters. While large models in general perform better, smaller, domain-adapted models are often competitive (Chalkidis et al., 2020; Lewis et al., 2020). The best hyper-parameter setting for fine-tuning often seems dependent on the end task. Especially for complex tasks, when more domain knowledge is needed, domain adaptation is beneficial (Chalkidis et al., 2020). Gu et al. (2021) state that mixed-domain pre-training is not necessary if sufficient domain-specific data is available for LM pre-training. However, in other studies it depends on the end-task if further pre-training or training from scratch is better. While empirical evidence on the best approach is not clear, resources available may limit the choice of the approach.

Gururangan et al. (2020) evaluate different adaption approaches in multiple domains, and examine further pre-training of LMs in low-resource scenarios. They distinguish between domain-adaptive pre-training (DAPT), and task-adaptive pre-training (TAPT), that is, pre-training on the (usually small amount of) unlabeled data of a given task. Combining the two approaches brings the best performance, but TAPT is often competitive with DAPT, while being much more resource-efficient. They further find that adaption to another domain is detrimental, and cross-task TAPT is not helpful. Thus, choosing the right data is crucial, but with this condition fulfilled, domain adaptation boosts performance also with limited data amounts.

3. Job advertisement corpora

We use two Swiss job ad datasets introduced in (Gnehm and Clematide, 2020). While both datasets are multilingual, roughly 80% are written in German. Due to the different size and collection method, each dataset has its own advantages for our experiments:

1. The Swiss Job Market Monitor (SJMM)[2] consists of representative yearly samples of job ads, from 1950 onwards up to now. This dataset is very well suited for longitudinal social science labor market research on the transformation of the labor market. For much information of interest to researchers, such as information about professions or skill requirements, there are human-generated class labels or textual annotations, which will serve us for the evaluation of language models in downstream NLP tasks.

[1] Data created for this paper is available via DOI 0.5281/zenodo.6497853, models via huggingface model hub.

[2] Available under https://www.swissubase.ch (Buchmann et al., 2021)

data source	# ads	# chars	size
SJMM corpus	80K	93M	92 MB
OA corpus	85K	150M	140 MB

Table 1: Training data per epoch for continued in-domain LM pre-training.

2. The online ad dataset (OA) is collected by a private company since 2012 through crawling of online job portals and company websites, and contains 2.25 million German-speaking job ads. In the context of neural language models, this is not huge data, but it is very valuable when it comes to adapting pre-trained models to the domain of job ads.

This domain is affected by strong data shift over time. Job ads are a fast evolving text type, and data shift relates to both changes in societal and labor market structure as well as changes in publication media and communication style. We observe an annual replacement rate of about 1% for the 1000 most frequent tokens (after removing stop words and punctuation), resulting in a high-frequency vocabulary overlap of less than 45% between 1990 and 2020.[3] To ensure a valid analysis for the entire period of interest from 1990 to the present as well as the near future, the impact of data shift on IE performance must be examined and addressed.

4. Experiments

4.1. Continued pre-training of LMs

General-domain pre-trained models: For continued in-domain pre-training of LMs, we use two BERT base models[4] (Devlin et al., 2019) trained on German data. Both models, bert-base-german-cased[5] – referred to as BERT-de in the following – and gbert-base (Chan et al., 2020) – GBERT in the following – improved the state-of-the art performance of base language models for German at the time of their release. The newer GBERT owes its competitive edge to both 10 times more training data (160 GB vs 12 GB) and whole word masking as part of the training objective. With these two starting point models, we can first examine whether the performance advantage of GBERT on the general domain also holds for our special domain, and second, whether the potential performance gains from continued in-domain pre-training for the two models are comparable, considering the larger difference in exposition to general domain text material. The commonalities and differences in the domain adaptation of BERT-de to jobBERT-de, and GBERT to jobGBERT are described below.

Domain vocabulary: As we continue to pre-train existing models, we are bound by the base vocabulary of those models and must ensure that our domain texts are optimally represented by that base vocabulary. First, we apply *character normalization* to minimize the number of words that are mapped to the <unk> token. Job ads use a variety of symbols for list items, most of which are not included in the base model vocabulary. Thus, these variations are all mapped to the most common form contained in the base vocabulary. Second, we apply *domain vocabulary insertion* to ensure that frequent domain-specific words get a good vector representation. We build our additional domain-specific vocabulary with SentencePiece (Kudo and Richardson, 2018) and fill the 3k reserved empty spots of BERT-de's 30k vocabulary with the most frequent subtokens. We insert subtokens, e.g., '#diplom' (*diploma*), regular words like 'Muttersprache' (*first language*), and common abbreviations like 'SAP'. In the GBERT model, unfortunately, only 100 entries are free in the 31.1k vocabulary. Given this small, negligible share of free spots, we leave the base vocabulary in this model as is. Regarding the LM vocabulary, we thus experiment with two different conditions for domain-adaptation provided by BERT-de versus GBERT.

Domain corpora: As for the further training of our two LMs in the job ad domain, we need to consider the peculiarities and different roles of our in-domain corpora. The large amount of data in the OA corpus seems very valuable for an effective adaptation. However, since the smaller SJMM corpus is important for social science analyses because it is both longitudinal and representative, our LMs should compute good representations of the job ads for this corpus. For this reason, we apply the following data weighting scheme: SJMM job ads are up-sampled so that we have around 3k ads per year from 1990 onwards, resulting in a total of 80k ads. OA job ads are down-sampled by a factor of 1/25, resulting in a total of 85k ads. We train for 25 epochs on a combination of these two datasets, using a different 1/25th of the OA corpus in each epoch (see Table 1). In this way, we aim for a good representation of SJMM ads going back to 1990, while taking advantage of the variation that comes with the amount of data in the more recent OA corpus.

Continued in-domain pre-training: We continue pre-training of the general-domain pre-trained BERT-de and GBERT on our job ads corpora with the masked language modeling task, resulting in two domain-adapted LMs, jobBERT-de and jobGBERT. To this end, we use the Huggingface Transformers library (Wolf et al., 2020) and follow largely its default training hyper-parameters. We train with a maximum sequence length of 512 subwords, start with a learning rate of 5E-05, use a linear learning rate scheduler with a batch size of 256 over 25 epochs, which results in 16.1k steps. In training of jobGBERT we follow Chan et al. (2020) and use a smaller batch size of 128 and a smaller initial learning rate of 1E-05. We train on NVIDIA Tesla T4 with 16 GB RAM for approximately one week.

[3] For the top 5k most frequent tokens the respective overlap is 35%, for top 10k it is 37%.

[4] 12 layers, hidden size of 768, 12 attention heads, 110M parameters.

[5] https://huggingface.co/bert-base-german-cased

job ad text	... we are looking for an experienced pharmaceutical assistant as sales consultant ..
profession	32 - Medical, pharmaceutical professions
main task	7 - customer service, sales, cashier
experience	1 - needed

Table 2: Example of job ad classification into profession, main job task and experience requirements

4.2. Fine-Tuning and evaluation on downstream NLP tasks

We fine-tune and evaluate four LM variants, BERT-de, GBERT, jobBERT-de and jobGBERT on different domain-specific tasks. First, we include document classification in our evaluation, and second, text zoning (a sequence labeling task), both in a classic supervised machine learning setting. For text zoning and one of the classification tasks, there are baselines by Gnehm and Clematide (2020) available, which we use for comparison. In text zoning, we further assess the impact of data shift over time on performance for most recent job ad data, as well as the mitigating effect of small amounts of labeled task data from this most recent time period.

Third, in order to evaluate the effect of domain-adaption on the performance of LMs on small datasets, we report data size ablation results for the ICT term recognition task in the job ads.

During continued in-domain pre-training of our LMs, we evaluated performance at different checkpoints – after every 5th epoch – on each task. While performance differences between the checkpoints are small, both in the case of jobBERT-de and jobGBERT, the checkpoint after epoch 20 reached the highest performance in the majority of the tasks. Therefore, we chose this checkpoint for the following, more comprehensive experiments.

For all evaluation tasks, we select the model reaching the highest accuracy on the dev set and evaluate on the test set. As recommended in the literature (Reimers and Gurevych, 2017), we repeat experiments and report performance estimates over three or five runs.

4.2.1. Document classification

Task: We assess the performance on job ad classification on three tasks, which are illustrated with an example in Table 2: Profession classification (34 classes) – this task is equivalent to the one in (Gnehm and Clematide, 2020), main task of the open position (21 classes), and required experience of the candidate (3 classes). For all three tasks, we have 25k labeled job ads available, which we split into 80% for training set and 10% each for the dev and test set.

Model architecture and training: The classification models use the last embedding layers of the transformer and fine-tune these in training. The '[CLS]' token is used to extract document embeddings, and fed into a linear layer on top to calculate class labels. We use the implementation by the Flair library (Akbik et al., 2018). Training of these models is done in batches of 16 and takes 5 epochs using Adam optimizer with a learning rate of 3.00E-05.

Results overview: Figure 1 presents evaluation results for all models on the three classification tasks. All three tasks deal with highly imbalanced classes. Since prediction quality for all classes is equally important to us independent of their frequency, we report accuracy and balanced accuracy, which corresponds to macro-recall in the multi-class case (Grandini et al., 2020). We observe that domain-adapted LMs outperform their general-domain starting point models in all three tasks, in accuracy and balanced accuracy. Furthermore, GBERT, which outperforms BERT-de on general-domain tasks (Chan et al., 2020), also achieves consistently better results than BERT-de in our domain-specific classification task. Consequently, the three models GBERT, jobBERT-de and jobGBERT reach higher accuracy in all tasks than BERT-de, and thus outperform the baseline for profession classification of 0.778 with BERT-de in Gnehm and Clematide (2020). Combining the better starting point model with continued in-domain training, jobGBERT proves to be the best performing model for our purposes, with balanced accuracy of 0.845 for experience, 0.734 for main job task, and 0.757 for profession classification.

High efficiency of in-domain training: Interestingly, in two of three classification tasks, jobBERT-de performs better than GBERT. This is impressive, considering the fact that GBERT was trained on almost ten times as much data as jobBERT-de, including whole word masking in the training objective. This result points at the importance of finding well-suited data for training domain- or applications-specific LMs. Moreover, it illustrates that even if resources – data itself or computational resources – are limited, it is possible to build well-performing domain-specific LMs.

Domain-adaptation and task complexity: Domain adaptation is beneficial for all classification tasks, but there are differences which might be related to task difficulty. The performance boost is largest for the main job task classification, with 2.1 point difference in balanced accuracy between jobBERT-de and BERT-de, and 1.4 points between jobGBERT and GBERT. For the other two classification tasks, the maximum improvement is 1.1 points each. According to both human judgement and mean model performance, the main job task classification appears to be the most difficult one.[6] Thus, as observed by Chalkidis et al. (2020), it seems that domain adaptation pays off especially for more complex tasks that require more in-domain knowledge.

Smaller gain of in-domain training for GBERT :

[6]Inter-annotator agreement measured by Krippendorff's alpha is 0.93 for experience requirements, 0.75 for main task, and 0.75 for a similar, but more-fine-grained profession classification task.

Figure 1: Evaluation of text classification tasks for all models on test set. Averages over 5 runs are indicated by numbers and a vertical black line in each bar. Darker colors show the range from min. to max. values over 5 runs.

Through all classification tasks, continued domain-specific pre-training proves beneficial, but the gain compared to the starting point model is smaller for jobGBERT than for jobBERT-de. The positive effects of further training on job ads may be smaller for GBERT for several reasons: first, we did not apply domain vocabulary insertion, and a less fitting vocabulary might limit the learning of good vector representations; second, the ratio of additional in-domain data vs. general-domain data is considerably smaller, meaning there is still a bigger bias towards the general domain; or third, learning effects are smaller because exposure to new, unseen material in continued training is lower in comparison. However, in line with findings by Gururangan et al. (2020), we conclude that very well performing general-domain LMs can still be further improved with relatively little amounts of well-chosen data.

4.2.2. Text zoning

Task: The second evaluation task, text zoning, refers to segmenting job ad texts into eight zones which differ regarding their content, e.g. job description, company description or skill requirements (Gnehm and Clematide, 2020). Since information on different topics can be densely packed in single sentences in job ads, text zoning is implemented as token level sequence labeling (for zone definitions see Table 3 and for an example of a zoned job ad see Figure 2).

Zone	Definition
z1	company description
z2	reason of vacancy
z3	administration & residual text
z4	job agency description
z5	material incentives
z6	job description
z7	required hard skills
z8	required personality (soft skills)

Table 3: Definitions of text zones (Gnehm and Clematide, 2020)

This task has been introduced in Gnehm and Clematide (2020) and for comparability, we use the same data, including job ads up to the year 2014 (n=22.5k in train, n=650 in dev and test set each). In addition, we aim at assessing the impact of data shift over time on zone tagging performance. To this end, we have newer labeled data available, covering the time period from 2015 to 2021 (n=350 job ads). We use 150 ads from this more recent time period as a second test set. In an additional experiment, we examine the effect of including small numbers of newer data when fine-tuning, by adding each of the remaining 200 ads three times to the original training data (n=23.1k job ads).

For/z1 our/z1 attractive/z1 product/z1 portfolio/z1 we/z3 are/z3 looking/z3 for/z3 an/z6 interior/z6 designer/z6 ./z6 You/z3 offer/z3 :/z3 -/z7 solid/z7 vocational/z7 training/z7 and/z7 experience/z7 ,/z7 -/z8 creativity/z8 and/z8 versatility/z8 ,/z8 -/z8 ideally/z8 you/z8 are/z8 between/z8 25/z8 and/z8 40/z8 years/z8 old/z8 ./z8 We/z3 offer/z3 :/z3 -/z6 a/z6 high/z6 degree/z6 of/z6 autonomy/z6 ,z6 -/z6 a/z6 large/z6 studio/z6 ,/z6 -/z6 an/z6 interesting/z6 and/z6 stimulating/z6 permanent/z6 position/z6 ./z6 Please/z3 send/z3 your/z3 application/z3 to/z3 POC/z3 ,/z3 ADDR/z3 ,/z3 Foto/z1 Hobby/z1 Inc./z1

Figure 2: Example of job ad with text zoning annotation (Gnehm and Clematide, 2020), translated from German to English

Model architecture and training: As for the classification models, we train sequence labeling models by taking representations from the last layer of the transformer, adding a linear layer on top, and fine-tuning on this task. To get token-level predictions, a pooling strategy for subword pieces is needed: The embeddings of the first subword is taken as the representation of the token. We use the implementation of Schweter and Akbik (2020) and rely largely on their training parameter recommendations: We use the AdamW optimizer and a once-cycle LR scheduler with an initial learning rate of 5e-06. We train for 20 epochs with batch size of 32.

Results overview: The evaluation of this task shows a similar picture to the evaluation of the classification

Model	Accuracy orig. train, orig. test set	orig. train, new test set	updated train, new test set
BERT-de	0.908	0.904	0.911
jobBERT-de	0.915	**0.934**	**0.944**
GBERT	0.913	0.919	0.925
jobGBERT	**0.916**	0.922	0.932
Model	**Balanced Accuracy** orig. train, orig. test set	orig. train, new test set	updated train, new test set
BERT-de	0.855	0.826	0.847
jobBERT-de	0.874	**0.894**	**0.941**
GBERT	0.869	0.859	0.885
jobGBERT	**0.880**	0.876	0.896

Table 4: Evaluation of text zoning for all models in 3 different settings. Reported are averages of 3 runs. For all models s.d.<0.0035 for accuracy, s.d.<0.015 for balanced accuracy. Best result per setting and evaluation measure in bold.

tasks before. Column 1 in Table 4 reports results for settings using the same data for training and evaluation as Gnehm and Clematide (2020). GBERT is a better general-domain starting model than BERT-de (balanced accuracy 0.869 vs. 0.855). Domain adaptation improves both starting models, such that the best performing jobGBERT reaches accuracy of 0.916 and balanced accuracy of 0.880, outperforming the baseline accuracy of Gnehm and Clematide (2020) by 0.6 points. The difference in pre-training in combination with domain adaptation results in a good margin of 2.5 points in balanced accuracy between this model and the worst performing BERT-de.[7]

Impact of data shift over time: An interesting question is, how the different models deal with the considerable data shift over time in the domain of job ads. If we evaluate on a more recent test set containing job ads from 2015 to 2021 (column 2 in Table 4), performance of BERT-de drops quite strongly by 3 points balanced accuracy, whereas jobBERT-de even reaches 2 points higher balanced accuracy. Since this test data was created by post-correcting data that was pre-annotated by jobBERT-de, this finding has to be treated with caution. Nonetheless, we assume the competitive edge of jobBERT-de over BERT-de is bigger than a supposed post-correction bias. Both evaluation results of GBERT and jobGBERT on newer test data are comparable to results on older data. These results imply that more extensive general LM pre-training as well as relatively limited continued in-domain training help mitigate the impact of data shift over time in downstream tasks.

Updating fine-tuning training data: Including newer data in model fine-tuning improves performance on new test data for all models, but we observe an interesting interaction effect with prior in-domain pre-training (see column 3 vs. column 2 in Table 4: Including new data in fine-tuning of domain-adapted models reduces the error rate by 1 point in the case of jobBERT-de, and by 1.1 points for jobGBERT. For the models with only general domain pre-training, the error rate is reduced by 0.7 for BERT-de, respectively 0.6 points for GBERT. Thus, it is especially the domain-adapted models that can benefit from updating end-task training data with small amounts of newer data. By combining domain adaptation and updating fine-tuned models with small amounts of new data, we can reduce the error rate from 9.6% (BERT-de, column 2), to 5.6% (jobBERT-de, column 3). This corresponds to a relative error reduction of more than 40%. The same procedure applied on the better initial model GBERT reduces error rate from 8.1% to 6.8% (jobGBERT column 3), still resulting in a relative error reduction of more than 15%. In sum, especially the combination of our two chosen strategies prove to be very effective to handle data shift over time.

Effect of vocabulary insertion: In text zoning too, continued pre-training on in-domain data leads to a larger improvement for jobBERT-de than for jobGBERT, in all three experimental settings (see Table 4). As mentioned before, this may have different reasons. In an in-depth analysis, we examined to what extent this difference is related to the domain vocabulary insertion technique we applied for jobBERT-de. We observed, e.g. in the third setting that accuracy for the 15% of tokens affected by vocabulary insertion was improved by 3.5% , whereas accuracy for the other 85% of tokens was improved by 3.7% through domain adaptation. Thus, the larger gain of in-domain training for jobBERT-de is not mainly caused by our vocabulary insertion technique. But interestingly, in the case of jobGBERT, improvement of accuracy for domain tokens was only 0.5%, for other tokens 1.1%. The large difference in performance gain for the respective tokens between jobBERT-de and jobGBERT shows that domain vocabulary insertion is a useful technique to learn vector representations of tokens that are important to the domain.

Impact of domain adaptation on minority zone classes: Domain adaptation has a stronger effect on balanced accuracy than on accuracy in all four settings reported in Table 4. On average, the relative improvement in balanced accuracy compared to initial models is 4.1%, in accuracy 2.1% for jobBERT-de, for jobGBERT it is 1.4% compared to 0.4%. This implies that domain adaptation is mainly improving performance, especially recall, for minority classes (see per-class results in Table 5). This makes sense because there are fewer training examples available for minority classes, and therefore better pre-trained textual representations are more helpful. This shows once again that domain adaptation is particularly valuable for more complex tasks.

[7] BERT-de reaches still 1.2 points higher accuracy than the same embeddings in Gnehm and Clematide (2020), due to a different model architecture and/or training parameters.

zone	Frequency abs.	rel.	precision	recall	F1
z1	22672	17.2%	0.911	0.913	0.912
z2	639	0.5%	0.822	0.768	0.795
z3	33186	25.2%	0.940	0.917	0.928
z4	964	0.7%	0.760	0.860	0.807
z5	2199	1.7%	0.839	0.851	0.845
z6	42610	32.4%	0.919	0.929	0.924
z7	16767	12.7%	0.921	0.935	0.928
z8	12515	9.5%	0.877	0.869	0.873

Table 5: Per zone frequencies, precision, recall and F1-values for best text zoning model jobBERT-de on original test set, reaching accuracy of 0.916, balanced accuracy of 0.880

4.2.3. ICT term recognition

Task: The last evaluation task deals with the detection of ICT concepts in job ads. ICT terms refer to single- and multi-word expressions from information and communication technology. It includes very general expressions like ”Computer”, widely known tools like ”MS-Office Programme” (*MS Office programs*) but also more specific tools and expressions like ”3D-CAD Systeme” (*3D CAD systems*) or ”embedded software engineer”. This term segmentation and classification task is approached as a named entity recognition (NER) type problem by applying a transition based NER method provided by spaCy[8].

Compared to the other tasks, only a small amount of training material is available. It consists of 2000 labeled job ads, split into 80% training set, and 10% for dev and test set each. The labeled data was created in an efficient iterative process using prodigy[9], an annotation tool for creating training data for machine learning models. The core idea of our approach was to train an initial model based on a small, manually annotated sample that helped to shape the annotation guidelines, and then iteratively increase the annotated material by correcting the models' predictions on new data samples with *prodigy ner.correct*. The present dataset was created in five rounds, with a total correction effort of 22 hours. Table 6 shows an example of a translated job ad with annotated ICT terms.

In order to cover as broad a spectrum of ICT terms as possible despite the relatively small amount of training material, a targeted sampling strategy was adopted. A MALLET topic model based on Latent Dirichlet allocation (LDA) (McCallum, 2002; Blei et al., 2003) computed on the entire job ad corpus with 100 topics has been used first to identify ads with an strong ICT focus. Later, by also including less ICT-oriented ads, we ensured that the model cannot only process ads from the ICT sector, but also learns to deal with ads that contain no ICT terms, only a few general ICT terms, or very profession-specific ICT terms. Overall, the final sample consists of one-third each of very ICT-heavy ads, ads with a moderate ICT focus, and ads evenly sampled from all 100 topics. In order to speed up the process even more, we did not annotate the entire job ads, but only segments from the text zones *z6: job description* and *z7: required hard skills* relevant to our research questions.[10]

Several years of professional experience in a similar function, very good **PC skills** (**MS-Office** , especially **Word** and **Excel** , if possible, experience with **Abacus**) as well as stylistically confident written and spoken German are required.

Table 6: Example of job ad segment with ICT term annotation

For our experiments, the samples have been stratified by years, since ICT terms have evolved considerably over the past 30 years. While there are only 96 different normalized ICT terms in 1990, there are over 3800 different such terms in 2020. In addition, a clear shift regarding ICT vocabulary can be observed in the entire corpus. Only 28% of the ICT terms from 1990 are still in the top 1k terms in 2020. And between the top 1k terms of the years 2015 and 2020, the agreement is not higher than 76%. This illustrates the rapid change to which the ICT vocabulary is subject.

Model architecture and training: We bootstrapped a transformer-based NER pipeline provided by spaCy. Based on the pre-trained transformer LMs, context-sensitive input representations are computed and used as features for the downstream NER component, which builds on a transition-based parser model[11] as described by (Lample et al., 2016). Apart from experimenting with different LMs, the settings and hyper-parameters used in the present study are based on the default settings of the spaCy pipeline based on the *spacy-transformers.TransformerModel.v1* and *spacy.TransitionBasedParser.v2* architectures.

Results overview: As shown in Figure 3, the two domain-adapted models also outperform the base models in ICT term recognition. With an F1 score of 0.890, the domain-adapted jobGBERT model achieves the best results overall. The fact that domain adaptation pays off is supported by the fact, that also jobBERT-de exceeds not only its base model BERT-de but also GBERT, which is pre-trained on significantly more material. Similarly to the previous tasks, the positive effect of domain adaptation, however, is higher for the BERT-de (+2.03 points) than for the GBERT (+0.96 points) base model.

Effect of training data size: To evaluate how performance is affected by the amount of training ma-

[8]https://spacy.io

[9]https://prodi.gy

[10]A buffer of maximum 10 tokens from other zones was allowed to keep the texts coherent.

[11]https://spacy.io/api/architectures#TransitionBasedParser

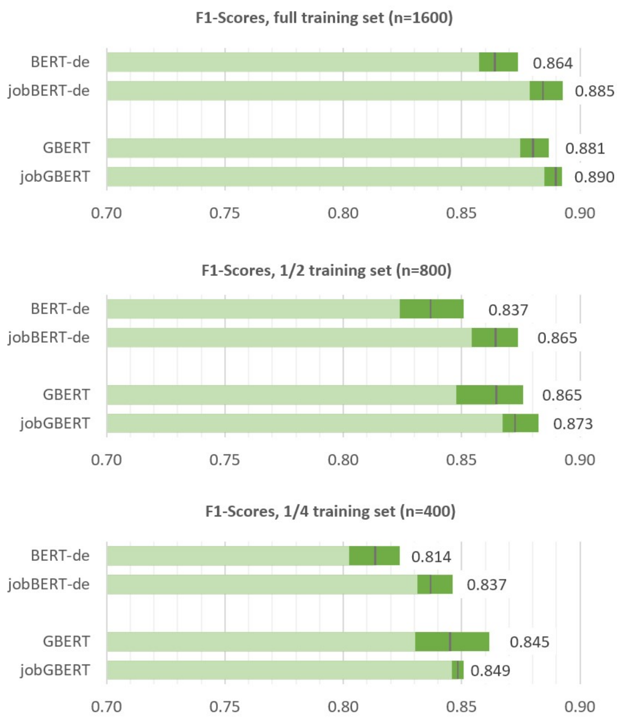

Figure 3: Evaluation of ICT term recognition task for models with varying training set sizes. Averages over 5 runs are indicated by numbers and a vertical black line in each bar. Darker colors show the range from min. to max. values over 5 runs.

terial, two additional model types were created with one-half (n=800) and one-quarter of the training material (n=400). For the BERT-de model, the effect of the domain adaptation becomes more pronounced with smaller training sets. Domain adaptation seems to help to deal with small amounts of data. Interestingly, the opposite phenomenon can be observed with the GBERT model: Here, the positive effect of domain adaptation gets smaller with less training material. In general, however, the variance of the five different runs in GBERT is relatively large, an effect that is reinforced by the training set reduction: For the smallest training set (n=400), the performance of GBERT fluctuates with F-scores between 0.830 and 0.862, whereas the results of the domain-adapted jobGBERT are more stable. Overall, also for the smaller amount of data, jobGBERT still is the best performing model. However, GBERT is almost as good and beats the domain-adapted jobBERT-de. Hence, the pre-training on the significantly larger database of GBERT compared to BERT-de pays off, especially when dealing with small datasets.

5. Conclusion

Domain adaptation techniques are beneficial over all different tasks and experimental settings, and bring relative error reductions of up to 5-8% in classification tasks, up to 15% in ICT term recognition, and up to 30% in text zoning. In line with the results of Chalkidis et al. (2020) our results suggest that domain adaptation especially pays off for more difficult tasks, e.g., classification of the main job task, or the recognition of minority classes.

The major competitive edge of GBERT over BERT-de in general domain NLP tasks is confirmed in all our domain-specific evaluation tasks. But our domain adaptation techniques are proving effective, and in most tasks, our domain adaptation of BERT-de, jobBERT-de, is competitive, or even better than the general-domain GBERT.

Improvement by domain-adaption is bigger for jobBERT-de than for jobGBERT in all experiments. This seems at least partly related to more extensive general pre-training of GBERT, leading to relatively larger exposition of this model to general domain texts, as well as smaller learning effects from additional in-domain data. However, our analysis showed that our domain vocabulary insertion technique for jobBERT-de, while not leading to major overall performance gains, still enables us to learn good representations of important domain-specific tokens.

Both more extensive general LM pre-training, as in the case of GBERT, and our relatively limited continued in-domain training mitigate the impact of data shift over time. Domain-adapted models can further benefit more from an update of labeled end-task training sets with small amounts of most recent data. Hence, continued pre-training of LMs with in-domain text seems especially advantageous in context of data shift over time.

Hyper-parameter grid search during task-specific fine-tuning has been shown to be helpful in other studies (Chalkidis et al., 2020). This strategy or ensemble solutions to boost the performance of our domain-specific text mining models remain for future work.

6. Acknowledgements

This work is supported by the Swiss National Science Foundation under grant number 407740 187333.

7. Bibliographical References

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual String Embeddings for Sequence Labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Alrowili, S. and Shanker, V. (2021). BioM-Transformers: Building Large Biomedical Language Models with BERT, ALBERT and ELECTRA. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 221–227, Online. Association for Computational Linguistics.

Beltagy, I., Lo, K., and Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3615–3620, Hong Kong, China, November. Association for Computational Linguistics.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Buchmann, M., Buchs, H., Busch, F., Gnehm, A.-S., Klarer, U., Müller, J., Müller, M., Sacchi, S., Salvisbert, A., and von Ow, A. (2021). *Stellenmarkt-Monitor Schweiz 1950 – 2020*. Soziologisches Institut der Universität Zürich.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. (2020). LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Chan, B., Schweter, S., and Möller, T. (2020). German's Next Language Model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Gnehm, A.-S. and Clematide, S. (2020). Text Zoning and Classification for Job Advertisements in German, French and English. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 83–93, Online, November. Association for Computational Linguistics.

Grandini, M., Bagli, E., and Visani, G. (2020). Metrics for Multi-Class Classification: an Overview. *arXiv:2008.05756 [cs, stat]*, August. arXiv: 2008.05756.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., and Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July. Association for Computational Linguistics.

Hedderich, M. A., Lange, L., Adel, H., Strötgen, J., and Klakow, D. (2021). A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June. Association for Computational Linguistics.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2019). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, page btz682, September.

Lewis, P., Ott, M., Du, J., and Stoyanov, V. (2020). Pretrained Language Models for Biomedical and Clinical Tasks: Understanding and Extending the State-of-the-Art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157, Online. Association for Computational Linguistics.

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In Yoshua Bengio et al., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2017). Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Ruder, S. (2019). *Neural transfer learning for natural language processing*. Ph.D. thesis, NUI Galway.

Schweter, S. and Akbik, A. (2020). Flert: Document-level features for named entity recognition.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (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, October. Association for Computational Linguistics.