Discover Artificial Intelligence

Discover

Research

Automated occupation coding with hierarchical features: a data-centric approach to classification with pre-trained language models

Parisa Safikhani[1] · Hayastan Avetisyan[1] · Dennis Föste-Eggers[1] · David Broneske[1]

Received: 22 November 2022 / Accepted: 28 January 2023
Published online: 13 February 2023

Abstract
Occupation coding is the classification of information on occupation that is collected in the context of demographic variables. Occupation coding is an important, but a tedious task for researchers in social science and official statistics that calls for automation. Due to the complexity of the task, currently, researchers carry out hand-coding or computer-assisted coding. However, we argue that, with the rise of transformer-based language models, hand-coding can be displaced by models, such as BERT or GPT3. Hence, we compare these models with state-of-the-art encoding approaches, showing that language models have a clear advantage in Cohen's kappa compared to related approaches, but also allow for flexible fine-grained coding of single digits. Taking into consideration the hierarchical structure of the occupational group, we also develop an approach that achieves better performance for the classification of different single digit combinations.

Keywords BERT · GPT3 · Occupation coding

1 Introduction

In empirical educational research or official statistics, occupations are central to many studies on social status or inequality (e.g., [1–5]). For example, information on parental occupations is often used to determine the socioeconomic status of parents. However, information on occupational activity is not only suitable for mapping aspects of social origin; a variety of occupation-related measures are now available that can be used to quantify occupation-specific health risks, gender segregation, or occupational closure [6].

Typically, the collection of occupational data in both written and online surveys is (still) carried out by means of open-ended questions [7, 8]. While job titles are usually quite short (mostly only a few keywords), some surveys collect more detailed job descriptions and activities, which could help in the post-processing of the job titles. The post-processing is a classification task of the textual descriptions using a chosen classification system. There are several classification systems, but all of them have hundreds of occupational codes, and the codes are always nested in hierarchies. The two central standard German categorization schemes are the German Classification of Occupations 2010 (KldB 2010) [9] and

Parisa Safikhani and Hayastan Avetisyan contributed equally to this work

✉ Parisa Safikhani, safikhani@dzhw.eu; Hayastan Avetisyan, avetisyan@dzhw.eu; Dennis Föste-Eggers, foeste@dzhw.eu; David Broneske, broneske@dzhw.eu | [1]German Centre for Higher Education Research and Science Studies (DZHW), Lange Laube 12, 30159 Hannover, Lower Saxony, Germany.

the International Standard Classification of Occupations 2008 (ISCO-08) [10]. While the KldB 2010 classifies specific job titles, ISCO classifies occupations. Our study leverages the former one because we want to classify German occupations.

Occupation coding is a complex activity, mostly manual or semi-automated, where tools assist human labelers in their decision-making of the thousands or hundred thousands of free-text answers of participants. Hence, an automation would reduce human effort enormously. The task of automating occupation coding can be seen as a subset of *automated text classification* (ATC), which is a well-described task. Within ATC, occupation coding is mostly related to *automated survey coding* (ASC), which also operates on short texts for social-science research and where accuracy is principal [11]. Current approaches in occupation coding, e.g., by Schierholz and Schonlau [12], use word similarity to train a classifier. However, they do not take into account the semantic connection between the job descriptions and job activities. Furthermore, they lack the ability to encode only a subset of the hierarchical digits of the KldB. As a solution, we argue that recently introduced pre-trained language models have the potential to solve this problem, leading to a boost in accuracy of automated occupation coding.

Hence, in this work, we fine-tune BERT and GPT3 for the task of automated occupation coding given an extensive, pre-labeled set of job occupations and activities from the DZHW Graduate Survey Series[1] and DZHW Survey Series of School Leavers.[2] It is a complex and challenging dataset since participants (adolescents) have only a vague idea of their future jobs and the dataset has not been extensively curated. Furthermore, due to the short texts, it is a challenge for language models due to missing linguistic features. Still, our approaches show a performance increase of 15.72 percentage points compared to the state-of-the-art methods. In summary, we contribute the following:

- We analyze the use case of occupation coding based on the current research by extracting important properties and requirements from the classification system and the scientists.
- We propose the use of transformer-based language models due to their superiority in extracting context, which is an important property for occupation coding.
- In our extensive evaluation, we show that the chosen language models outperform the state-of-the-art in occupation coding and even have a significantly better performance for a fine-grained encoding of single digits of the KldB.

To the best of our knowledge, this is the first attempt of automating (German) occupation coding using pre-trained language models, such as BERT and GPT3.

The remainder is structured as follows. In Sect. 2, we present the most important research in automated occupation coding and, in Sect. 3, we formally define our classification task and the pre-trained models, which we use for classifying the occupations. We also introduce our data and the classification system KldB 2010. Section 4 describes the experiments and discusses evaluations of the adopted approaches to our classification task. Section 5 discusses open questions, while Sect. 6 concludes the work and presents opportunities for further research in applications to occupational coding and short text classification with hierarchical labels.

2 Related work

Coded occupation data usually represent the premise for any further analysis and studies [13] and the significance of automating the coding process has been emphasized by numerous researchers. More precisely, even if 35–50 percent of the occupations can be coded automatically, the time saved, in comparison to applying hand-coding, is enormous. Furthermore, if especially easy-to-code professions are automatically coded, this would significantly reduce the workload of coders [13]. Numerous attempts have been made to accomplish this goal.

An extensive body of research has been conducted by Schierholz [14], who introduced a method based on a supervised learning approach for automating the coding of occupational data using KldB 2010. The study concludes that the best results can be achieved by combining rule-based coding with supervised learning. Furthermore, Schierholz and Schonlau [12] compared seven occupation coding algorithms found in the literature and compared their performance on five datasets from Germany. The best results are obtained by leveraging Tree boosting (cod index), i.e., the list of job titles is merged with coded responses from previous surveys before using this combined training

[1] https://www.dzhw.eu/en/forschung/projekt?pr_id=467.
[2] https://www.dzhw.eu/en/forschung/projekt?pr_id=465.

data for statistical learning. The performance rates of the four algorithms that rely on training data only (memory-based reasoning, adapted nearest neighbor, multinomial regression, tree boosting (XGBoost)) are closely comparable within each data set.

Three other approaches for automated occupation coding are proposed by Gweon et al. [15]: (1) a combination of two statistical learning models for different levels of aggregation, (2) a combination of a duplicate-based approach with a statistical learning one, and (3) a modified nearest neighbor approach. Using data from the German General Social Survey (ALLBUS), the two best-performing methods turned out to be the modified nearest neighbor method (NN-3) and a hybrid method (hybrid-3/4digit), which substantially improved the accuracy compared with both statistical learning by itself and the duplicate method at any production rate in the ALLBUS data. As the percentage of duplicates decreases, NN-3 gains a relative advantage over the hybrid method.

Lim et al. [16] proposed a system without index database and achieved higher performance using KoBERT. Results of 95.65 percent, 91.51 percent, and 97.66 percent were obtained for occupation/industry and industry code classification on standardized texts.

Furthermore, Decorte et al. [17] proposed a neural representation model for job titles, called JobBERT, by supplementing the pre-trained language model with co-occurrence information from skill labels extracted from job postings. The model leads to significant improvements over the use of generic sentence encoders for the title normalization work task, for which we publish a new evaluation benchmark. The method is based on the premise that skills are the essential components that define a job.

The study of Bao et al. [18] introduced a strict algorithm that will be able to identify the NOC (2016) codes using job titles and industry information as input exclusively. The ACA-NOC was applied to over 500 manually-coded job and industry titles. The accuracy rate at the four-digit NOC code level was 58.7 percent and improved when broader job categories were considered (65.0 percent at the three-digit NOC code level, 72.3 percent at the two-digit NOC code level, and 81.6 percent at the one-digit NOC code level). Several different search strategies were employed in the ACA-NOC algorithm to find the best match, including exact search, minor exact search, like search, near (same order) search, near (different order) search, any search, and weak match search. In addition, a filtering step based on the hierarchical structure of the NOC data was applied to the algorithm to select the best matching codes. Garcia et al. [19] designed and tested an automated coding prototype classification system ENENOC (the ENsemble Encoder for the National Occupational Classification), encompassing several steps: data cleaning, exact match search, multi classifier ensembling, hierarchical classification, and multiple output selection. The prototype was benchmarked on a manually annotated data set comprising 64,000 records. It produced a top-1 Per-Digit Macro F1-Score of 0.65 and a top-5 Per-Digit Macro F1-Score of 0.76. In the absence of exact matching between job title input and NOC category descriptions, the input data is embedded using the TF-IDF algorithm and Doc2Vec. The embeddings are fed into a hierarchical ensemble classifier that uses classical machine learning techniques: Random Forests, Support Vector Machine, and K-Nearest Neighbor.

Savic et al. [20] developed a web tool named Procode to code free texts against classifications and recoding between different categories. To this end, three text classifiers—Complement Naïve Bayes (CNB), Support Vector Machine (SVM), and Random Forest Classifier (RFC)—were investigated using k-fold cross-validation. 30,000 free texts with manually assigned classification codes of the French classification of occupations (PCS) and French classification of activities (NAF) were available. For recoding, Procode integrated a workflow that converts codes of one classification to another according to existing crosswalks. CNB resulted in the best performance among the three investigated text classifiers, where the classifier accurately predicted 57–81 percent and 63–83 percent classification codes for PCS and NAF, respectively. SVM led to lower results (by 1–2 percent), while RFC coded accurately up to 30 percent of the data. There are already some approaches to occupation coding, mostly simple ML approaches based on statistical similarity measures. However, the results while applying those methods are not satisfactory enough. Although language models have been used in this area, they were not intended for classification based on categorization, but rather for skill to job offer coding. Therefore, for the task of occupation coding, we want to use robust language models such as BERT and GPT3 to achieve better results.

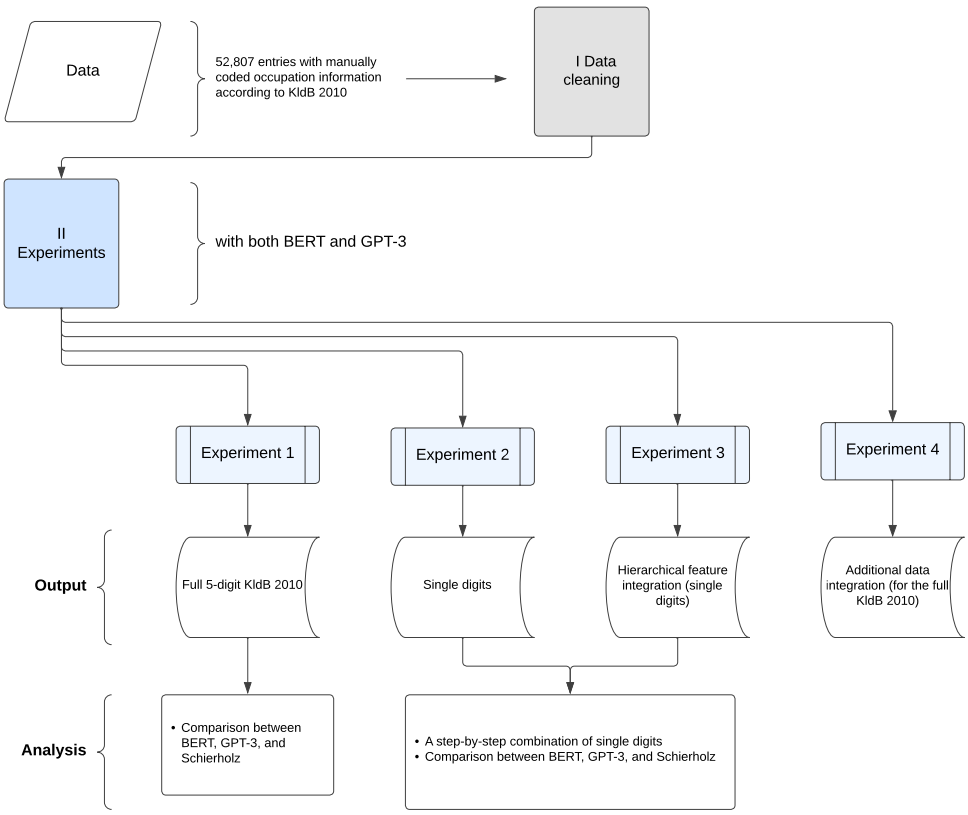

Fig. 1 Study process

3 Methods

In our study, we utilize occupation coding to meet the specific needs of the researchers. As we rely on the KldB classification system, we will present it in detail. Additionally, we have a requirement to code a subset of the digits within the KldB, and will discuss the models we have chosen to accomplish this. To provide clarity, we also provide an outline of our study regarding the methods and the experiments (see Fig. 1).

3.1 KldB 2010

The German *classification of occupations* (KldB 2010) [9] is developed by the German Federal Employment Agency and is valid since January 1, 2011. A transformation from KldB-2010 to the International Standard Classification of Occupations 2008 (ISCO-08) is possible through conversion keys, which is used by several researchers [21].

The KldB is a five-digit, hierarchically structured code, as shown in Table 1. The first digit of the code denotes the occupational area (Berufsbereich) like "Agriculture, forestry, animal husbandry and horticulture" or "Health, social affairs, teaching and education", which is more and more specified by the following three digits (the second digit denotes occupational main group (Berufshauptgruppe), the third digit denotes the occupational group (Berufsgruppe), and the fourth digit denotes the occupational sub-group (Berufsuntergruppe)). The fifth digit, occupational type (Berufsgattung), denotes the requirement level (Anforderungsniveau) or degree of complexity of the occupational activity like "Helper/apprentice occupations" or "professionally oriented activities".

A challenge of the KldB 2010 is that it contains an enormous number of classes, i.e., 1286 occupational categories when using all five digits. However, the KldB 2010 encodes two dimensions, occupational group and requirement level, which we can exploit when optimizing the training of the language models.

Table 1 German classification of occupations: structure and differentiation possibilities. Adapted from the German Classification of Occupations 2010 [9]

Basic structure			Differentiation options			
Level	Outline level	Number of classes	Assistant	Skilled worker	Specialist	Expert
1st level	Occupational area (Berufsbereich)	10	9	10	10	10
2nd level	Occupational main group (Berufshauptgruppe)	37	26	37	37	35
3th level	Occupational group (Berufsgruppe)	144	54	125	129	123
4th level	Occupational sub-group (Berufsuntergruppe)	700	60	414	442	370
5th level	Occupational types (Berufsgattung)	1286	60	414	442	370

Table 2 Example of hierarchical description of KldB directory, starting with first digit of code one

Code	Hierarchy description
1	Occupations in agriculture, forestry, farming, and gardening
11	Occupations in agriculture, forestry, and farming
111	Occupations in farming
1110	Occupations in farming (without specialization)
11101	Occupations in farming (without specialization)-unskilled/semi-skilled tasks
11102	Occupations in farming (without specialization)-skilled tasks
11103	Occupations in farming (without specialization)-complex tasks
11104	Occupations in farming (without specialization)-highly complex tasks

3.1.1 Occupational group

The horizontal differentiation of an occupation group is coded in positions 1 to 4, and the level of differentiation increases with each position, which results in 700 classes. The group names and descriptions are an essential source of information for any trained model. Besides numerical codes for the groups, there are hierarchical descriptions of combinations of digits from left to right. The hierarchical description of code 1 is shown in Table 2. The description of the first digit shows us that the occupations that have the number one as the first digit belong to the occupations in the field of agriculture, forestry, farming, and gardening. If the second digit is also one, the description shows that these occupations can no longer be categorized under the group gardening. If the third digit is also one, it shows that the occupations with the first three one digits are categorized only under farming. The description of the combination of the whole digits shows the complexity level of the occupations categorized under farming. These eight hierarchical descriptions can be used as features to categorize the occupations for this example under the 4 existing KldB numbers, 11101, 11102, 11103 or 11104.

3.1.2 Requirement level

The requirement level is coded on the 5th digit with overall four classes, shown in the right part of Table 1. The number of differentiation options indicates how many entries would result if the entries of the outline level were each broken down by requirement level. For example, since there are no assistants or experts for some occupational subgroups, these figures cannot simply be derived from the number of entries per outline level.

Due to the two dimensions and as a result of the hierarchical structure of the first dimension, exploiting the relation between the first four digits and splitting the coding into separate models is a reasonable optimization approach for us.

3.2 The case for single-digit coding

The classification of occupations is imperative in official statistics and in social science research. However, not all studies use the KldB codes to their full extent. Numerous studies in different disciplines leverage separate digits of the KldB codes in their analyses. For instance, some studies use the whole KldB number as in [22] and others leverage separate digits of the KldB codes in their analyses [23–27]. This ranges from use cases where only the first two or three digits are used [23] to a combination of the first two or three digits with the fifth digit [24, 25]. Currently, in those studies, the whole 5-digit KldB is encoded and, afterward, necessary digits are extracted for further analysis. However, due to the big number of classes for a 5-digit KldB encoding, an approach that only encodes the necessary number of digits could be worthwhile.

Hence, we pose the questions, (1) whether we could exploit the relation between single digits for occupation coding and (2) whether we can fine-tune the task of occupation coding for use cases where only a subset of the KldB digits is needed.

3.3 Used models

As mentioned in related work, the only approach to automated German occupation coding using artificial intelligence was by Schierholz and Schonlau [12], who used supervised learning algorithms that nowadays do not have the best performance for multi-class text classification. Training deep language models on our data is time-consuming and computationally expensive. Pre-trained language models are, therefore, attractive because they reduce the burden on practitioners to provide the appropriate resources (time, hardware, and data) to train the models [28, 29]. Furthermore, occupations are mostly short texts, and short text classification is difficult to model statistically due to fewer features and training data. As shown in Luo and Wang [30] using a pre-trained language model can be useful for this task.

In search of a better and more time-efficient performance for automatic coding of occupations, we propose to use two pre-trained language models BERT [31] and GPT3 [32] for classification of occupations with the whole KldB number as well as for single digits of the KldB. Due to the fact that occupation texts are not sentences, there are not many semantic and syntactic relations. Thus, we think that GPT3, which has a capacity of 175 billion parameters and has the syntactic ability to assign words, could be useful for our task. From the other side, we have additional data, that can represent semantic relationships between digits. In using this information as features, a bidirectional model like BERT with its masking capability could play a better role.

3.3.1 BERT

BERT is a transformer-based language model that stands for Bidirectional Encoder Representation. It is the first deeply bidirectional, unsupervised language representation. With just one additional output layer, this pre-trained model can be fine-tuned to create state-of-the-art models for a large variety of tasks such as text classification, question answering, and language inference. BERT is trained with the masked language modeling (MLM) task, in which some tokens in a text sequence are randomly masked. Then, the masked tokens are independently recovered by conditional encoding vectors obtained by a bidirectional transformer [31]. For German-speaking applications, such as occupation coding, using a German-language BERT model is useful, e.g., the one built by deepset.[3] Google Multilingual BERT also supports the German language, but the German BERT model significantly outperforms Google's multilingual BERT model [33], which is why we chose deepset's model for our experiment.

3.3.2 GPT3

GPT3 is the third generation of auto-regressive transformer-based language models. GPT1 was built with unsupervised pre-training and supervised fine-tuning for a single task. GPT1 is followed by GPT2, which did not need supervised fine-tuning for a specific task and was well suited for multiple tasks. The GPT3 model, having a similar architecture basis as previous models, is able to make accurate predictions without gradient updating or fine-tuning [32, 34]. The depth of GPT3 has increased considerably. It has by far more parameters than BERT [31, 35].

[3] https://www.deepset.ai/.

GPT3 is unidirectional, which is its biggest limitation compared to BERT. The choice of architectures during fine-tuning is limited by being unidirectional. In GPT3, a left-to-right architecture was used, where each token can only respond to previous tokens in the self-attention layers of the transformer [32, 34]. It means, the word can be predicted by GPT3 based on previous predictions [31, 35]. Another disadvantage of GPT3 is that it has no ability to understand the semantics and context of the query, but only a statistical ability to match words [36].

4 Experiments

Even the best machine learning algorithms cannot perform well without carefully collected and reprocessed data. Recently, data-centric AI [37] has gained prominence. Its main goal is to improve not the training algorithm for the model, but the data preprocessing for model accuracy. Hence, we first present how we cleaned and prepared our data. Afterward, we describe our experiments. The experiments are:

1. Our first experiment analyzes the performance of the pre-trained language models for the whole five-digit KldB. As a result, we can draw conclusions about whether our approaches can outperform state-of-the-art coding schemes for German occupations.
2. The second experiment tests the performance for use cases that require only a subset of the available KldB digits (see Sect. 3.2). Thus, we expect our models to improve due to the smaller number of classes.
3. In the third experiment, we investigate whether the second experiment can be improved by propagating decisions from previous predictions. This way, we target to exploit the hierarchical nature of the KldB system for both training and testing the models to improve our predictions.
4. The last experiment investigates the influence of integrating entire hierarchical features on prediction performance for whole five-digits KldB.

4.1 Data cleaning

Our data originates from the DZHW Graduate Survey Series and the DZHW Survey Series of School Leavers conducted throughout the years 2005–2015 as paper surveys that were digitalized afterward. Occupations are gathered using a free-text field for the participants to fill in. To facilitate the subsequent coding, there is another free-text field, where participants should enter typical activities in their jobs. Hence, both data—*job titles* and *typical tasks*—can be used in our study as input for classifying the exact occupation. Therefore, the usually short texts of job titles, which are challenging for language models due to their limited context, are extended with more verbose task descriptions, if they were provided.

To have a ground-truth, the data set consisting of 52,807 entries was encoded by several trained coders using a standardized set of rules. 10,000 entries of the dataset have additional information about professional tasks. Hence, we base our study on a quite reliable, but not perfect dataset. In fact, we found 935 entries that were not assigned a correct KldB number and removed those from our dataset. This improved our overall prediction performance by 2–3 percentage points in the following experiments. Due to the amount of our data, the old heuristic of a 70%/30% split of training and testing is not the optimal option in this case [38]. Therefore, in order to as efficiently as possible leverage the existing data for training, we allocated 90% for the training and 10% for the validation. As a test set, we used a separate test set with 4329 entries, which is being classified after saving the respective trained model.

4.2 Experiment 1: classifying occupations on full five-digit KldB

In our first experiment, we classify the occupations (job titles and, if available, their tasks) as a multi-class text classification task on the whole 5-digit KldB. This experiment should evaluate whether the baseline performance of BERT and GPT3 can keep up with the algorithms of [12] for a challenging dataset with limited semantics and short texts. To this end, we fine-tuned the models as follows:

Table 3 Performance of BERT, GPT3 and Schierholz on the same test set

Model	BERT	GPT3	Schierholz
Cohens' kappa [%]	64.22	29.93	48.50

Table 4 Performance of BERT and GPT3 fine-tuned with separate single digits of KldB, compared to split digits from whole KldB numbers from Schierholz and Schonlau [12]

Label	BERT [%]	GPT3 [%]	Schierholz [%]
1st digit	84.49	75.97	72.69
2nd digit	80.52	70.16	68.83
3rd digit	76.93	64.43	60.14
4th digit	70.76	63.70	52.49
5th digit	69.80	57.78	53.32

1. Fine-tuning BERT: For our experiments, we used the pre-trained language model German BERT.[4] The model is implemented with Pytorch[5] in Google Colab with GPU acceleration. We achieved the best results with seven epochs, a learning rate of 1e−5, and a batch size of 16, which covered the size of all occupations plus their tasks. Moreover, we used the Adam optimizer.
2. Fine-tuning GPT3: For fine-tuning the encoder of GPT3, we followed the approach proposed by Cansen Çağlayan.[6] Our implementation is written in TensorFlow.[7] We fine-tuned GPT3 with 92,316,156 trainable parameters. The Adam optimizer was used here.

The calculated Cohen's kappa on our test set, which is coded by both fine-tuned models, can be seen in Table 2. We have also coded our test set with Schierholz' model as the state-of-the-art approach.[8]

Table 3 shows that the results of BERT are superior to the other models. It outperforms Schierholz' algorithm by 15.72 percent points and GPT3 by 34.29 percent points. Surprisingly, although our data consists of short sequences and do not carry a lot of semantic and syntactic information, GPT3 did not achieve higher performance. Possibly, the reason might be the fact that our data has a high number of labels, which might be too much for GPT3. On the other hand, the results show that despite short text and not enough semantic information between words in our data, BERT could still capture the semantic relations between classes of KldB.

4.3 Experiment 2: predicting single digits of KldB

Considering the fact that the number of labels has a significant influence on the performance of the pre-trained model, we split the KldB numbers into five digits. Instead of fine-tuning a model with the entire KldB numbers with 1286 different classes, we fine-tuned five models, each with a single digit of KldB numbers, each comprising at most ten different labels. The results can be seen in Table 4. The Cohen's kappa values refer to the performance of our model on the test set.

As expected, the number of labels has a large impact on the behavior of BERT and GPT3. Compared to Experiment 1, the accuracy values for both models have increased substantially. Interestingly, the performance of GPT3 is in the same range as BERT's, which suggests that GPT3's poor performance in Experiment 1 is due to the number of classes.

A second insight is that the first dimension (occupational group) has a direct impact on the performance of the models. While the first three digits only have an accuracy difference between 2–3 percentage points, the gap is slightly bigger for predicting the fourth digit. The decreasing accuracy manifests that especially the fourth digit is harder to predict, since textual differences inside the class members with the same prefix are only minor and, thus, harder to learn.

Since most studies use several concatenated digits, our approach allows combining the predicted digits without restriction. Hence, we assembled the digits step by step from left to right and calculated the resulting Cohen's kappa, which is shown in

[4] https://huggingface.co/bert-base-german-cased.
[5] https://github.com/pytorch/pytorch.
[6] https://github.com/kmkarakaya/Deep-Learning-Tutorials/blob/master/Multi_Class_Text_Classification_End_to_End_Example_.ipynb.
[7] https://www.tensorflow.org/api_docs/python/tf.
[8] https://github.com/malsch/occupationCoding.

Table 5 Cohen's kappa of assembled digits, predicted by GPT3 and BERT with separate digits, compared with Schierholz

Model	First two digits	First three digits	First four digits	Whole digits
BERT [%]	78.90	73.16	63.82	56.14
GPT3 [%]	70.04	63.60	55.37	49.82
Schierholz [%]	70.28	63.23	54.45	48.05

Table 5. Overall, BERT is still the best approach due to its better performance for single-digit classification. It is striking that the combination of 5 digits in GPT3 achieves a better Cohen's kappa value than when trained with the whole KldB number.

4.4 Experiment 3: hierarchical feature integration in training and testing

Based on the bias-variance analysis in conjunction with the training loss and validation loss values, we found that our model is under-fitted. From a data-centric AI perspective, we further investigated the problem and came up with the idea that we can use the available KldB 2010 information as features, which are shown in Table 2. In the KldB 2010, each digit of labels has a literal description, which is a subcategory of the previous digit and a concise description for the next digit.

To this end, we improved the fine-tuning procedure for the models at the second to fifth levels by incorporating additional input features. Specifically, before fine-tuning the second model for the second digit of the label, we added the literal description of the first digit as a feature to the input. We then repeated this step for the remaining digits, using literal descriptions of the combination of the first, second, third, and fourth digits as additional features so that if X_i be the input data (job descriptions) of *i*th entry from our dataset *D* and $L_{i,j}$ are the literal descriptions when $j \in \{1, 2, 3, 4\}$, as follows:

$L_{i,1}$ = the literal description of the first digit

$L_{i,2}$ = the literal description of the combination of the first and second digit

$L_{i,3}$ = the literal description of the combination of the first, second, and third digit

$L_{i,4}$ = the literal description of the combination of the first, second, third, and fourth digit

and $k \in \{1, 2, 3, 4, 5\}$ be the digit number of KldB. For instance, the fifth model was fine-tuned with $F_{i,5}$ input which represents the job description including the feature set for the fifth digit, as follows:

$$F_{i,5} = X_i + L_{i,1} + L_{i,2} + L_{i,3} + L_{i,4}$$

For a generalizable fine-tuned model, the validation and test sets should look as similar as possible [38]. Therefore, we built our test set for this experiment step by step, similar to the validation and training set. After we coded the test set with the fine-tuned model, which is trained with the first digit as the label, we added the literal information of the first digit as strings, which was predicted, to the occupations and made a new test set. To code the test set for the second digit, we used the new test set. We repeated this step again for the further digits so that the entire process can be presented as follows:

Let $Y_{i,k}$ be the output label of *i*th entry and the *k*th digit (digits in KldB 2010), $L_{i,j}$ as described be the literal descriptions of *i*th entry, M_k be the fine-tuned model for digit *k*, *V* be the validation set and T_k be the test set for digit *k*. The process for fine-tuning the models for the second to fifth level can be represented as:

```
For k = 1 to 5:
      Fine-tune M_k on F_k and Y_k using V
      Code T_k with M_k
      Add L_{i,(k-1)} to T_k as features
```

This process aims to improve the model's generalizability by gradually incorporating the literal descriptions of the digits as features and building the test set in a similar way to the validation and training set.

We report the performance of our fine-tuned model on the stepwise hierarchically built test set in Table 6. Although the results of BERT for single digits did not improve compared to the last experiment (cf. Table 4 vs. Table 6), considering the relationship between digits led to an increase of 4.46 percentage points in mean Cohen's kappa when combining the digits (see Table 7).

Table 6 Performance of the fine-tuned BERT models, which are fine-tuned with literal information of KldB numbers as features

Training set		Cohen's kappa [%]	
Text = Occupation +.	Label	BERT	GPT3
–	First digit	84.59	75.97
Literal description of the first digit	Second digit	80.34	70.79
Literal description of the first and second digit	Third digit	77	66.35
Literal description of the first, second, and third digit	Fourth digit	69.10	59.36
Literal description of the first, second, third, and fourth digit	Fifth digit	69.64	59.23

Table 7 Cohen's kappa of assembled digits, after they are being coded with the fine-tuned models with separate digits as labels and with step-wise literal description of KldB 2010 as features

Model	First two digits	First three digits	First four digits	Whole digits
BERT [%]	81.08	77.18	68.72	62.17
GPT3 [%]	69.81	64.18	56.57	51.64

4.5 Experiment 4: fine-tuning BERT and GPT3 using additional data for encoding the whole KldB

Our goal was to reduce bias, but at the same time not contribute to over-fitting. Hence, we use the existing information from KldB while making sure that our test and training set do not look too different. To this end, we created a new data set. Half of our data had as input occupations plus the literal term for the first digit plus literal terms for the combination of first two, first three, first four, and total KldB digits. The other half had just the occupation as input. Both half parts had the whole KldB number as label. With this experiment, we could increase the Cohen's kappa of BERT to 66.01% and the Cohen's kappa of GPT3 to 42.60%, which is a slight but significant increase compared to the results in Table 3.

5 Discussion

The number of occupations to be encoded and the quality of the encoded occupations depend heavily on the quality of the answers collected. The more complete and accurate the answers are, the better they can be coded (either manually or automatically). Spelling mistakes and abbreviations, prevalent in web surveys, also lead to significantly poorer auto-coding results. Therefore, it is worth doing a rough cleaning of the data [13]. When coding occupations, it is helpful to draw on additional information that enables the occupational activity to be classified as precisely as possible. Without this supplementary information, some details cannot be coded entirely [13]. Furthermore, our results can be applied not only in the context of German occupational coding, but also on a larger scale, i.e., internationally.

In the future, we would like to develop a recommendation for job encoding by suggesting the 3 codes with the highest softmax value to the user and letting them decide.

6 Conclusion

Occupation coding is an important task, which transforms textual job descriptions and task to a common numeric scheme. For our use case of complex German questionnaire data, we choose the KldB 2010 classification system. It gives us the possibility to not only encode the full five-digit KldB number, but being able to also return a prefix/subset of the five digits. To this end, we have presented two approaches for occupation coding using the two pre-trained language models, BERT and GPT3. Overall, BERT was able to outperform the state-of-the-art approach by 15.72 percent points on the whole KldB number. Furthermore, we were able to increase Cohen's kappa beyond the value of predicting the whole KldB number due to our tuning for single KldB digits with enhanced hierarchical information.

Author contributions The contributions of the authors were distributed in the following way: Conceptualization, PS, HA, DF, DB; methodology, PS, HA, DB; model implementation, PS, HA; validation, PS, HA; writing—original draft preparation, PS, HA; writing—review and editing, PS, HA, DB; supervision, DB. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Competing interests The authors declare no competing interests.

References

1. Fujishiro K, Xu J, Gong F. What does "occupation" represent as an indicator of socioeconomic status?: Exploring occupational prestige and health. Soc Sci Med. 2010;71(12):2100–7.
2. Connelly R, Gayle V, Lambert PS. A review of occupation-based social classifications for social survey research. Methodol Innov. 2016;9:2059799116638003.
3. Schooler C, Schoenbach C. Social class, occupational status, occupational self-direction, and job income: A cross-national examination. In: Sociological Forum, vol. 9, pp. 431–458 (1994). Springer.
4. Hatt PK. Occupation and social stratification. Am J Sociol. 1950;55(6):533–43.
5. Qi Y, Liang T, Ye H. Occupational status, working conditions, and health: evidence from the 2012 china labor force dynamics survey. J Chin Sociol. 2020;7(1):1–23.
6. Christoph B, Matthes B, Ebner C. Occupation-based measures-an overview and discussion. KZfSS Kölner Zeitschrift für Soziologie und Sozialpsychologie. 2020;72(1):41–78.
7. Peycheva DN, Sakshaug JW, Calderwood L. Occupation coding during the interview in a web-first sequential mixed-mode survey. J Off Stat. 2021;37(4):981–1007.
8. Rapley TJ. The art (fulness) of open-ended interviewing: some considerations on analysing interviews. Qual Res. 2001;1(3):303–23.
9. Klassifikation der Berufe K. Band 1: Systematischer und alphabetischer Teil mit Erläuterungen. Bundesagentur für Arbeit (2010)
10. Office IL. International Standard Classification of Occupations 2008 (ISCO-08): Structure, Group Definitions and Correspondence Tables. Geneva: International Labour Office; 2012.
11. Gendlin A, Viechnicki P. Computer-assisted historical occupation coding. URL: https://scholar.google.com/scholar?hl=de&as_sdt=0%2C5&q=+Gendlin+A%2C+Viechnicki+P.+Computer-assisted+historical+occupation+coding.&btnG=
12. Schierholz M, Schonlau M. Machine learning for occupation coding-a comparison study. J Surv Stat Methodol. 2021;9(5):1013–34.
13. Züll C. The coding of occupations. GESIS Survey Guidelines. 2016.
14. Schierholz M. Automating survey coding for occupation. PhD thesis; 2014.
15. Gweon H, Schonlau M, Kaczmirek L, Blohm M, Steiner S. Three methods for occupation coding based on statistical learning. J Off Stat. 2017;33(1):101–22.
16. Lim J, Moon H, Lee C, Woo C, Lim H. An automated industry and occupation coding system using deep learning. J Korea Converg Soc. 2021;12(4):23–30.
17. Decorte J-J, Van Hautte J, Demeester T, Develder C. Jobbert: Understanding job titles through skills. arXiv preprint arXiv:2109.09605. 2021.
18. Bao H, Baker CJ, Adisesh A, et al. Occupation coding of job titles: iterative development of an automated coding algorithm for the Canadian national occupation classification (aca-noc). JMIR Form Res. 2020;4(8):16422.
19. Garcia CAS, Adisesh A, Baker CJ. S-464 Automated Occupational Encoding to the Canadian National Occupation Classification using an Ensemble Classifier from TF-IDF and Doc2Vec Embeddings. London: BMJ Publishing Group Ltd; 2021.
20. Savic N, Bovio N, Gilbert F, Canu IG. Procode: the swiss multilingual solution for automatic coding and recoding of occupations and economic activities. arXiv preprint arXiv:2012.07521. 2020.
21. Tiemann M, Kaiser F. Klassifikationen der Berufe-Begriffliche Grundlagen, Vorgehensweise, Anwendungsfelder. na, Bonn; 2013.
22. Kraft MHG. How important are linguistic competencies on the german labour market? a qualitative content analysis of job advertisements. EJEBS. 2021;v5i3:35–41. https://doi.org/10.26417/ejes.
23. Geis-Thöne W. Zuwanderung hat den gesundheitsbereich gestärkt. Technical report, IW-Kurzbericht; 2020.
24. Koebe J, Samtleben C, Schrenker A, Zucco A. Systemically relevant but little recognized: Compensation of indispensable occupations underperformed in the Corona crisis. 2020. https://www.diw.de/de/diw_01.c.792754.de/publikationen/diw_aktuell/2020_0048/systemrelevant__aber_dennoch_kaum_anerkannt__entlohnung_unverzichtbarer_berufe_in_der_coronakrise_unterdurchschnittlich.html.
25. Guggemos J. Analyse beruflicher tätigkeitsfelder von wirtschaftspädagogen/-innen anhand von daten des karriereportals xing. Zeitschrift für Berufs-und Wirtschaftspädagogik. 2018;114(4):551–77.
26. Diel A. Ein viertel der pharmabeschäftigten arbeitet in der produktion. Technical report, IW-Kurzbericht; 2019.
27. Frank F, Jablotschkin M, Arthen T, Riedel A, Fangmeier T, Hölzel LP, Tebartz van Elst L. Education and employment status of adults with autism spectrum disorders in Germany–a cross-sectional-survey. BMC Psychiatry. 2018;18(1):1–10

28. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, et al. Language models are unsupervised multitask learners. OpenAI blog. 2019;1(8):9.
29. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21(140):1–67.
30. Luo L, Wang Y. Emotionx-hsu: Adopting pre-trained bert for emotion classification. arXiv preprint arXiv:1907.09669. 2019.
31. Devlin J, Chang M-W, Lee K, Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018.
32. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, et al. Language models are few-shot learners. Adv Neural Inf Process Syst. 2020;33:1877–901.
33. Chan B, Schweter S, Möller T. German's next language model. arXiv preprint arXiv:2010.10906. 2020.
34. Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training (2018)
35. Minaee S, Kalchbrenner N, Cambria E, Nikzad N, Chenaghlu M, Gao J. Deep learning-based text classification: a comprehensive review. ACM Computing Surveys (CSUR). 2021;54(3):1–40.
36. Floridi L, Chiriatti M. Gpt-3: its nature, scope, limits, and consequences. Minds Mach. 2020;30(4):681–94.
37. Ng A, Laird D, He L. Data-centric ai competition. DeepLearning AI. 2021. https://deeplearning-ai.github.io/data-centric-comp/. Accessed 9 Dec 2021.
38. Andrew Ng. Yearning for machine learning. 2018. https://info.deeplearning.ai/machine-learning-yearning-book.