University of Magdeburg

Faculty of Computer Science

Master's Thesis

Occupation coding using a pretrained language model by integrating domain knowledge

Author:

V.R Prasanth Vaidya Karanam

December 2, 2022

Advisors:

Prof. Dr. rer. nat. habil. Gunter Saake

Department of Databases and Software Engineering

Dr.-Ing. David Broneske, Hayastan Avetisyan, Parisa Safikhani

DZHW - German Centre for Higher Education Research and Science Studies

Karanam, V.R Prasanth Vaidya:
Occupation coding using a pretrained language model by integrating domain knowledge
Master's Thesis, University of Magdeburg, 2022.

Abstract

Generally, surveys are considered one of the popular mechanisms to collect data regarding a problem associated with any field. The researchers use the collected data to conduct studies or statistical analysis to identify the critical aspects of a problem associated with the concerned domain. The occupation data of the public is often used by researchers in scientific studies to understand the dynamics of the labor market and health risks associated with occupations and determine social status. Since the surveys consist of responses in textual form, occupation classification schemes are used to standardize the responses by assigning an occupation code. This enables the researchers to work with standardized responses to conduct further studies and statistical data analysis. However, assigning the occupation code to the free text responses using classification schemes is challenging due to the variation in the quality of the user responses and the various categories of occupation codes. Existing methods in the research address this problem by implementing classification rules from occupation classification schemes, using the classification scheme as an index, or using statistical techniques combined with Machine Learning to perform occupation coding. However, we observed that classification schemes were mainly used as an index or a look-up mechanism in the process of occupation coding. In addition, apart from the traditional Machine Learning methods, the application of pre-trained language models to this task is still being explored. Hence, we propose using the occupation information provided in the classification schemes as an input to pre-trained language model (BERT) for occupation coding. We extract the different types of information associated with the occupations from the official occupation classification scheme (KLDB) and provide it as input to BERT through classification tasks. This methodology was evaluated on the Deutsche Zentrum für Hochschul- und Wissenschaftsforschung (DZHW) occupation data for the occupation coding task. However, the proposed approach only showed a slight improvement in the classification performance with the integration of domain knowledge.

Acknowledgements

I want to take this opportunity to express my sincere gratitude to my supervisors at the University, Prof. Dr. Gunter Saake and Dr. David Broneske at DZHW, along with the mentors Hayastan Avetisyan and Parisa Safikhani, for providing me with a motivating topic related to my area of interest for my master thesis that helped me to acquire and develop new skills. I would like to thank them for their constant support and guidance throughout the course of my master thesis. Their expertise and feedback helped me to improve on my research methodology and kept me motivated.

I am extremely grateful to my parents, family, and friends for their admiration, encouragement, and support, which helped me to stay strong and focused throughout the journey toward my master's degree.

Contents

List of Figures

List of Tables

1. Introduction

Occupation is regarded as an important factor in the fields of Social Sciences, Economics, and Healthcare. In the context of Socio-Economic research, it is used in performing wage impact analysis [Ste11], observing job trends in the labor market, and determining socio-economic status [CGL16] of an individual since an occupation indicates the income and education qualifications. In the case of healthcare, researchers perform epidemiological studies [MK03, MCBB09] to understand the health risks and workplace hazards associated with an occupation. In order to do such assessments the data regarding occupations is collected from individuals through surveys. These surveys are designed to collect user data for research on aspects like employment, health, education, etc. The surveys consist of open-ended questions for respondents regarding their current or previous or aspiring occupation, skills, and educational qualifications required for their occupation. The respondents are entitled to answer similar questions in a survey about their elders or partners.

Table 1.1 consists of questions regarding the employment of an individual and their partner taken from The German General Social Survey (ALLBUS 2018) [fS19], which is the English translation for "Die Allgemeine Bevölkerungsumfrage der Sozialwissenschaften". For questions F060 and F071 in Table 1.1 the respondents are expected to describe their current or previous main occupation and the activities/skills involved in that corresponding role. In addition, the respondent is expected to answer if the occupation has a unique or different name. Also, the respondents can provide the details about their spouse (F081) or living partners (F094). In graduate surveys, the questions can be about the aspired occupation (Wunschberufe) or the jobs that the respondent has applied for (Ausbildungsberufe) (see Table 1.2). In Table 1.2, one can observe that the questions can also be about the occupation of respondents' parents.

As the questions are descriptive in nature, the responses consist of details about the occupation in the form of text with short sentences and sometimes keywords. In order to perform any further analysis on the occupations of the respondents, firstly, these responses have to be coded or converted into a standard format by assigning occupation codes or IDs to the user's responses based on official occupation cate-

F060 - Falls Befragter hauptberuflich erwerbstätig ist	**F081 - Falls zusammenlebender Ehepartner hauptberuflich erwerbstätig ist**
Welche berufliche Tätigkeit üben Sie in Ihrem Hauptberuf aus? Bitte beschreiben Sie mir Ihre berufliche Tätigkeit genau. _____ Hat dieser Beruf, diese Tätigkeit noch einen besonderen Namen? _____	Welche berufliche Tätigkeit übt Ihr(e) (Ehe)Partner(in) in seinem/ ihrem Hauptberuf aus? Bitte beschreiben Sie mir die berufliche Tätigkeit genau. _____ Hat dieser Beruf, diese Tätigkeit noch einen besonderen Namen? _____
F071 - Falls Befragter ehemals hauptberuflich erwerbstätig war	**F094 - Falls Lebenspartner des Befragten hauptberuflich erwerbstätig ist**
Welche berufliche Tätigkeit übten Sie in Ihrem Hauptberuf zuletzt aus? Bitte beschreiben Sie mir Ihre letzte berufliche Tätigkeit genau. _____ Hat dieser Beruf, diese Tätigkeit noch einen besonderen Namen? _____	Welche berufliche Tätigkeit übt Ihr Partner/ Ihre Partnerin in seinem/ ihrem Hauptberuf aus? Bitte beschreiben Sie mir die berufliche Tätigkeit genau. _____ Hat dieser Beruf, diese Tätigkeit noch einen besonderen Namen? _____

Table 1.1: Questions related to occupations - ALLBUS

gorization schemes. The task of categorizing or assigning an occupation ID to a user's occupation is termed as occupation coding. The occupation categorization schemes vary across countries. For example, the German Classification of Occupations (KLDB 2010) [PM13] is used in Germany and NOC [BBA20] in Canada. The occupation categorization schemes consist of hierarchically structured occupations with a unique identification number. For example, KLDB 2010 consists of a unique 5-digit number assigned as an ID to different occupation categories. The details of this scheme are further explained in the Section 2.1.2

1.1 Occupation Coding Methods - Overview

Occupation coding can be done through manual and (semi-) automatic coding techniques. The process of manually assigning an official occupation ID to the user's textual responses is an exhaustive task and requires the availability of domain experts and human coders. So, researchers developed computer-assisted and automatic

Ausbildungsberufe	Wunschberufe / Aspirationen
Welcher Ausbildungsberufe wird dies voraussichtlich sein? Nennen Sie bitte die beiden Ausbildungsberufe, für die Sie sich am häufigsten beworben haben.	Unabhängig von ihrer aktuellen Situation, welchen Beruf würden Sie später einmal am liebsten ergreifen? Und wenn Sie einmal an alles denken, was Sie jetzt wissen: welchen Beruf würden Sie wohl tatsächlich einmal ergreifen?
Ausgeübte Berufe	**Elternberufe**
Was war/ist ihr hauptsächlicher Tätigkeitsbereich? Welche beruflichen Aufgaben erfüllten Sie in diesem Tätigkeitsbereich?	Welchen Beruf üben bzw. übten Ihre Eltern beruflich aus? Vater:_____ Mutter: _____

Table 1.2: Questions related to occupations - DZHW

coding methods to fasten this process so that they can focus on performing statistical analysis and publishing key results for their respective problems. The computer-assisted coding technique suggests a list of possible occupation categories for a user's response, and the human coder assigns the best category from the list. This can be considered as a semi-automatic process for occupation coding. In the case of automatic coding techniques, there are rule-based [BBA20] methods, hybrid methods [BBA20], and Machine Learning [SS20] methods. The rule-based approaches rely upon a set of rules created by domain experts considering the occupation coding process. These rules are designed by considering the taxonomy of the official occupation schemes. The computer-assisted and automatic coding methods often use the occupation schemes as an index to perform text search and text matching to assign an occupation ID for a given user response.

In Machine Learning (ML) context, occupation coding is considered a text classification problem. The ML models assign an occupation ID for a user response. The models are trained on data consisting of occupation IDs assigned by experts to the user responses. Classifying or assigning an occupation ID to a user's responses is often tricky since the responses are often vague, with keywords and minimal details. In addition, as there are a higher number of target classes or occupation IDs, it is a high-dimensional problem, and the algorithms require quality training data to achieve a good classification performance. We discuss some widely known Machine Learning methods implemented in automatic and computer-assisted coding algorithms in the related work section.

1.2 Contribution & Research Questions

In this thesis, we propose an approach to impart the domain knowledge of an official occupation classification scheme into a pre-trained language model (BERT) and address the following research questions.

RQ1: How can domain knowledge be provided to BERT to address the scenario of short text classification in occupation coding data?

The existing Machine Learning methods (ML) train models on the user responses to perform occupation coding. The user responses often contain less amount of text which affects the performance of the model. Hence, we propose an approach to leverage the capability of BERT to utilize the domain knowledge of the occupation categorization scheme to overcome the problem of lack of text in user responses. The proposed approach consists of intermediate tasks to impart the domain knowledge into BERT.

RQ2: Does integrating domain knowledge from the intermediate tasks improve the performance of the BERT classifier in occupation coding?

In this research question, we evaluate whether the proposed approach helps in improving the performance of a BERT classifier compared to the baseline using accuracy as an evaluation metric.

RQ3: Does the performance of the classifier improve on augmenting domain data with the user responses during the training phase?

We combine the data of user responses and the official occupation categorization schemes for the training BERT and later compare the performance with a baseline to evaluate whether this affects the classifier's performance.

1.3 Structure of Thesis

We structured the thesis report into six chapters. In the first chapter, we discuss occupation coding and the contributions of this thesis. The second chapter provides background information related to occupation coding and transformer architectures. Further, in the third chapter, we discuss the approaches implemented to address occupation coding and how external knowledge is incorporated with BERT as part of the related work. In the fourth chapter, we discuss the concept and implementation of the proposed approach. We discuss the experiments and their results of the proposed approach in the fifth chapter. In the sixth chapter, we provide the conclusion and future work.

2. Background

The purpose of this chapter is to provide necessary background information about the thesis topic. First, we discuss official occupation classification schemes used in occupation coding, followed by transformer architectures BERT and SBERT.

2.1 Occupation Coding Schemes

As discussed in the previous Chapter 1, official occupation categorization schemes often regarded as taxonomies are used in the process of occupation coding. In this section, we discuss a couple of widely known categorization schemes namely ISCO - 08 and KLDB 2010.

2.1.1 ISCO - 08

The International Standard Classification of Occupations 2008, also known as ISCO-08, was published by the International Labour Organization (ILO). It was developed for a comparative analysis of occupations between countries in cross-cultural surveys. ISCO-08 is a 4-level hierarchical classification that classifies all the occupations in the world into 436 specific categories as mentioned in [ISC08]. The meaning of each hierarchy level and the number of occupation categories at each level are mentioned in the Table 2.1.

The ISCO-08 broadly classifies the different types of occupations into ten major groups that are mentioned in Table 2.2. It assigns a 4-digit unique ID (ISCO-08

Hierarchy Level	Meaning	Categories
1	Major group	10
2	Sub-major group	43
3	Minor group	130
4	Unit group or Line of work	436

Table 2.1: ISCO - 08 classification details

ISCO-ID	Major groups
1	Managers
2	Professionals
3	Technical and Associate Professionals
4	Clerical support workers
5	Service and sales workers
6	Skilled agricultural, forestry, and fishery workers
7	Craft and related trades workers
8	Plant and machine operators, and assemblers
9	Elementary occupations
0	Armed forces occupations

Table 2.2: 10 Major groups in ISCO - 08

ISCO-ID	Title
3	Technicians and Associate Professionals
31	Science and Engineering Associate Professionals
311	Physical and Engineering Science Technicians
3111	Chemical and Physical Science Technicians
3112	Civil Engineering Technicians
3113	Electrical Engineering Technicians
312	Mining, Manufacturing, and Construction Supervisors
3121	Mining Supervisors
3122	Manufacturing Supervisors
3123	Construction Supervisors
32	Health Associate Professionals
321	Medical and Pharmaceutical Technicians
3211	Medical Imaging and Therapeutic Equipment Technicians

Table 2.3: Major group 3 - Classification hierarchy from ISCO-08

code) indicating the 4-levels of classification. For example, 3111 is the ISCO-08 ID for Chemical and Physical Science Technicians. The significance of each digit for the code 3111 is as follows:

- Digit 1: 3 - Major Group
- Digit 2: 1 - Sub-major group
- Digit 3: 1 - Minor Group
- Digit 4: 1 - Line of work - Unit Group

Table 2.3 displays an extract from [ISC08], which shows some occupation groups in the classification hierarchy for the major group 3 - Technicians and Associate Professionals. As observed, the classification scheme segregates Engineering and Health Professionals using the subgroup (Digit 2) and groups the comparable occupations related to Engineering and Mining using the minor groups (Digit 3) 1 and 2. Finally, the actual occupation or the line of work is indicated by the unit group (Digit 4) e.g., 3111 Civil Engineering Technicians and 3113 Electric Engineering Technicians.

2.1.2 KLDB 2010

Initially, two German national classification schemes: Klassification der Berufe 1998, KLDB 1998 by Federal Employment Agency (Bundesagentur für Arbeit) and Klassifikation der Berufe 1992, KLDB 1992 by Federal Statistical Office were used for occupation coding till the year 2010. Because both classifications were derived from theoretical work in the 1960s, KLDB 2010 was introduced to replace the outdated classifications.

The KLDB 2010, which we refer to as domain data in this work, is a hierarchical classification of 5 levels. The interpretation of each hierarchy level and the number of occupation categories at each level are depicted in the Table 2.4. The hierarchy is similar to the ISCO-08 scheme but has an additional 5th level.

Hierarchy Level	Meaning	Categories
1	Occupational fields - Berufsbereiche	10
2	Main occupational groups - Berufshauptgruppen	37
3	Occupational groups - Berufsgruppen	144
4	Occupation subgroups - Berufsuntergruppen	700
5	Occupational categories - Berufsgattungen	1286

Table 2.4: KLDB 2010 - classification details

The 5th level as mentioned in [PM13], indicates Auxiliary and semiskilled occupations, specialized occupations, complex occupations for specialists, and highly complex occupations. The KLDB 2010 broadly categorizes all the occupations in Germany into ten main occupational fields (Berufsbereiche) mentioned in Table 2.5. It assigns a 5-digit unique ID (KLDB ID) to the occupations, where the first four digits indicate the professional specialization and the 5th digit indicates skill level. According to [PM13], there are four types of skill levels, each indicated by a digit as follows:

KLDB ID	Berufsbereich
1	Land-, Forst- und Tierwirtschaft und Gartenbau
2	Rohstoffgewinnung, Produktion und Fertigung
3	Bau, Architektur, Vermessung und Gebäudetechnik
4	Naturwissenschaft, Geografie und Informatik
5	Verkehr, Logistik, Schutz und Sicherheit
6	Kaufmännische Dienstleistungen, Warenhandel, Vertrieb, Hotel und Tourismus
7	Unternehmensorganisation, Buchhaltung, Recht und Verwaltung
8	Gesundheit, Soziales, Lehre und Erziehung
9	Sprach-, Literatur-, Geistes-, Gesellschafts- und Wirtschaftswissenschaften, Medien, Kunst, Kultur und Gestaltung
0	Militär

Table 2.5: KLDB 2010 - Berufsberieche

- 1 - Helfer-/Anlerntätigkeiten
- 2 - fachlich ausgerichtete Tätigkeiten
- 3 - komplexe Spezialistentätigkeiten
- 4 - hoch komplexe Tätigkeiten

Table 2.6 is an extract of the hierarchy from [PM13] for the occupational area "Rohstoffgewinnung, Produktion und Fertigung". It is evident that occupations with different professional specializations are denoted using a different KLDB ID e.g., 2111 and 2112. It also indicates that occupations with varying skill levels under a professional specialization 2111 are differentiated using the 5th digit in a KLDB ID e.g., Helfer-/Anlerntätigkeiten (1) and fachlich ausgerichtete Tätigkeiten (2).

2.2 Deep neural networks for NLP

Natural Language Processing (NLP) involves analyzing and representing human language through sophisticated computational algorithms involving Machine Learning and Deep Learning (DL). NLP deals with the task of analyzing the aspects like syntax and semantics of the human language. So it is used in various scenarios like Text generation and summarization, Question-Answering (QA), Text classification, Automatic speech recognition (ASR), etc. DL algorithms based on CNN [Kim], RNN [YKYS17], and Transformer [VSP+] architectures are often utilized to perform such complex tasks. The neural networks are the building blocks of these architectures. Neural networks were developed to model the human brain's computation mechanism, composed of neurons.

Figure 2.1 depicts a simple feed-forward neural network (FFNN) with input, hidden, and output layers. The direction of the arrows indicates the flow of information

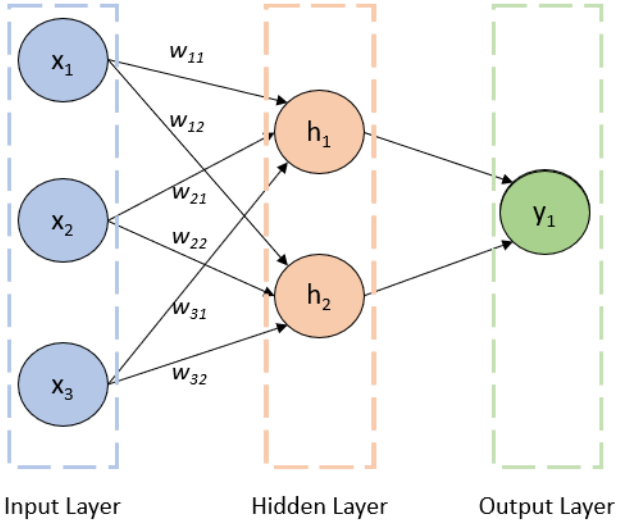

Figure 2.1: Example of feed-forward neural network

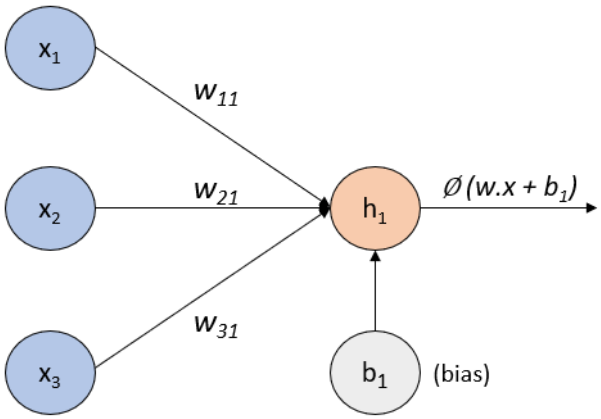

Figure 2.2: Detailed view of hidden layer neuron in feed-forward neural network

KLDB ID	Title
2	Rohstoffgewinnung, Produktion und Fertigung
21	Rohstoffgewinnung und -aufbereitung, Glas- und Keramikherstellung und -verarbeitung
211	Berg-, Tagebau und Sprengtechnik
2111	Berufe im Berg- und Tagebau
21111	Berufe im Berg- und Tagebau - Helfer-/Anlerntätigkeiten
21112	Berufe im Berg- und Tagebau - fachlich ausgerichtete Tätigkeiten
21113	Berufe im Berg- und Tagebau - komplexe Spezialistentätigkeiten
21114	Berufe im Berg- und Tagebau - hoch komplexe Tätigkeiten
2112	Berufe in der Sprengtechnik
21122	Berufe in der Sprengtechnik - fachlich ausgerichtete Tätigkeiten
21123	Berufe in der Sprengtechnik - komplexe Spezialistentätigkeiten
21124	Berufe in der Sprengtechnik - hoch komplexe Tätigkeiten
212	Naturstein- und Mineralaufbereitung und -verarbeitung und Baustoffherstellung
2120	Berufe in der Naturstein- und Mineralaufbereitung und -verarbeitung und Baustoffherstellung (ohne Spezialisierung)
21201	Berufe in der Naturstein- und Mineralaufbereitung und -verarbeitung und Baustoffherstellung (ohne Spezialisierung) - Helfer-/Anlerntätigkeiten

Table 2.6: Berufsbereich 2 - Classfication Hierarchy

in the network. The nodes in each layer are connected to all the nodes in the preceding and succeeding layers. The connections between the nodes have different weights, as shown in Figure 2.1. In addition, the nodes in hidden and output layers consist of non-linear activation functions that typically alter the input values to non-linear output. Sigmoid, ReLU, and Tanh are examples of some activation functions. Figure 2.2 indicates the detailed view of the operations that occur at a neuron in the hidden layer. The following equation in the Figure 2.2

$$h_1 = \phi(w.x + b_1) \tag{2.1}$$

where :

h_1 = hidden state
ϕ = activation function
w = Weight matrix
. = dot product
x = input matrix
b_1 = bias

indicates that dot product is performed on the inputs and weights, and bias is added before applying an activation function.

As it is observed that in feed-forward neural networks, the information flow is uni-directional, they do not recall the information they previously received. This is not

ideal for NLP applications since the human language or text input is a sequence of words of variable length. This makes the feed-forward architecture unsuitable for analyzing the syntax and semantics. To address these limitations, architectures like RNN and their variants LSTM were introduced. Figure 2.3 indicates the flow of information in an FFNN and RNN.

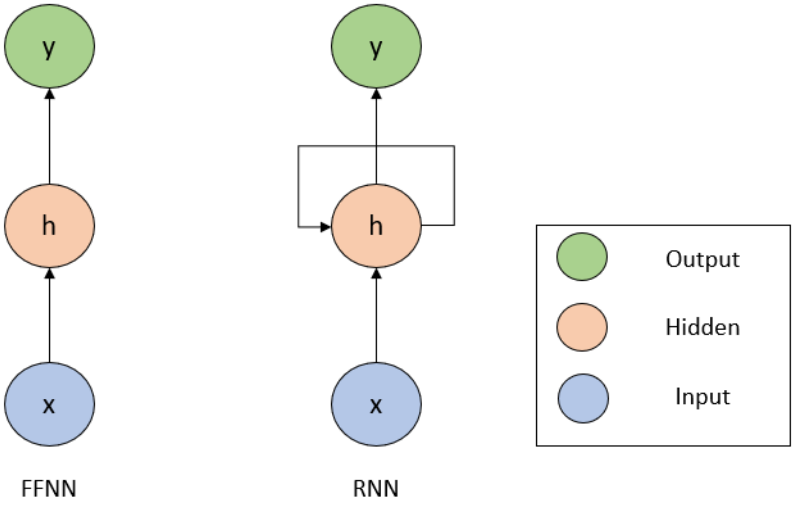

Figure 2.3: Comparison of the flow of information in FFNN and RNN

2.2.1 RNN and LSTM

Recurrent neural networks (RNN) are a variant of the artificial neural network often used to work with sequential data since they can maintain the previous input information to predict the output. This property of retaining past information makes them suitable for NLP tasks since the RNNs can maintain the contextual information of the text as the past information. Figure 2.4 depicts how RNN utilizes the information from the past through the weighted connections between the hidden states. It can be observed that for a current input x_t, the hidden node h_t uses the previous information from the node h_{t-1} and gives the output y_t. The following equations Equation 2.2 and Equation 2.3 indicate how the information from the previous state 't-1' is used as input to predict the output of a current state 't.'

$$h_t = \phi(w_x.x_t + w_h.h_{t-1} + b_h) \tag{2.2}$$

$$y_t = \phi(w_y.h_t + b_y) \tag{2.3}$$

where :

ϕ = activation function
w_x, w_y = Weight matrices
x_t = input at time step t
. = dot product
h_t, h_{t-1} = hidden state at time step t and t-1
b_h, b_y = bias at hidden node and output node

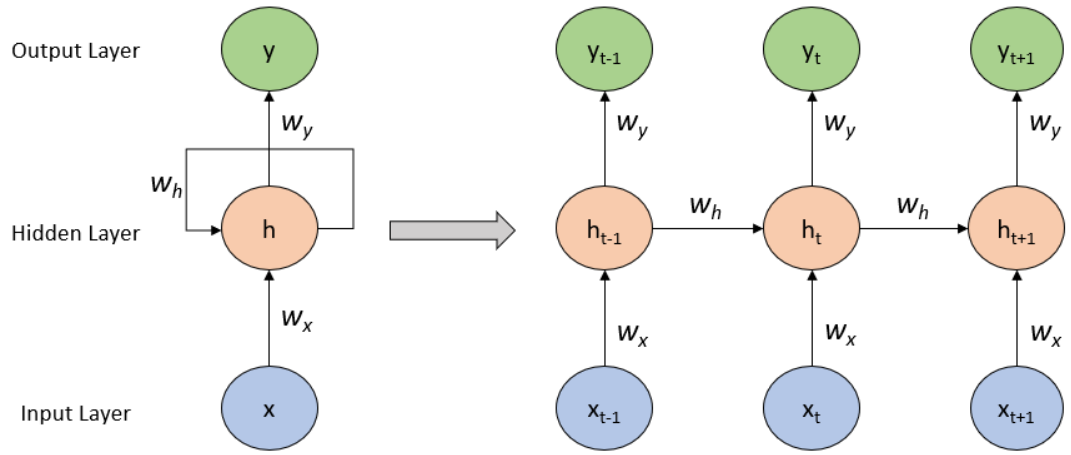

Figure 2.4: Unfolded view of an RNN. 't' indicates time step

In RNN, forward prorogation is performed through the operations mentioned in Equation 2.2 and Equation 2.3 to predict the output, and later the error is computed during the training phase. The neural networks rely on backpropagation [RHW86] to calculate gradients and update the weights after the error computation. Backpropagation, when performed in an RNN, is termed backpropagation through time (BPTT). As mentioned earlier, the RNNs capability to retain contextual information makes them suitable for NLP tasks, but they tend to retain such information and work well for shorter sequences. For an example sequence, *'In summer the temperatures are _____'* the prediction from RNN can be *'In summer the temperatures are high'*. Nevertheless, for a longer sequence like *'John won a national swimming competition five years ago before switching to a professional football career, but he still likes _____'*. The prediction, in this case, should be *'swimming'*. However, RNNs fail to work in this scenario since they are prone to the problem of exploding and vanishing gradient [PMB12] when the input sequence is extensive. Since the network is unrolled for various time steps, the same weights shared across time steps are updated using BPTT leading to the vanishing gradient problem. Long short-term memory (LSTM) networks [HS], a variant of RNN, were proposed to overcome the challenges of retaining the contextual information in larger sequences and exploding and vanishing gradients.

Unlike RNNs, LSTMs can retain the context in more extensive sequences since they can choose whether to forget or retain the historical information. The Figure 2.5 shows an unfolded view of the LSTM[1] with the mathematical operations occurring within an LSTM [2] cell. As it can be observed from the Figure 2.6 the LSTM consists of input, forget and output gates. The forget gate decides which information from the previous time steps will be used in the future time steps.

[1]https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[2]https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate-conceptual-guide-with-keras-2a650327e8f2

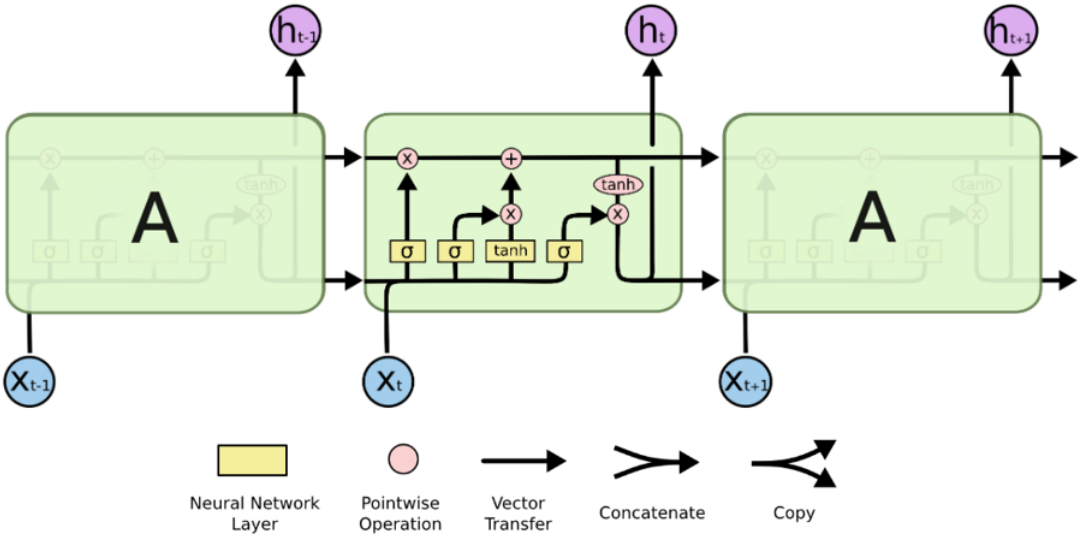

Figure 2.5: Unfolded view of LSTM

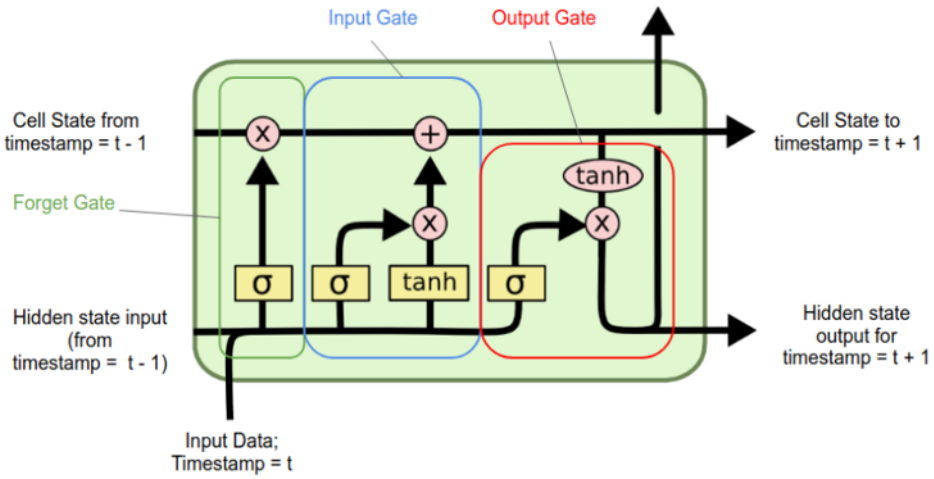

Figure 2.6: Detailed view of an LSTM cell

In the context of NLP, contextual information is important to understand the semantics and syntax of the text or language. RNNs based on encoder-decoder [BCB15] architectures capture such context information using a fixed length context vector as shown in Figure 2.7. However, this caused problems with longer sequences, as mentioned in [BCB15] since the information is compressed into a fixed length vector. The attention mechanism [VSP+] addressed this limitation by opting to create a sequence of context vectors and selectively use them. This eventually led to the usage of attention mechanism in NLP tasks like text summarization, text classification, and sentiment analysis as mentioned in [GLT21, LPGL].

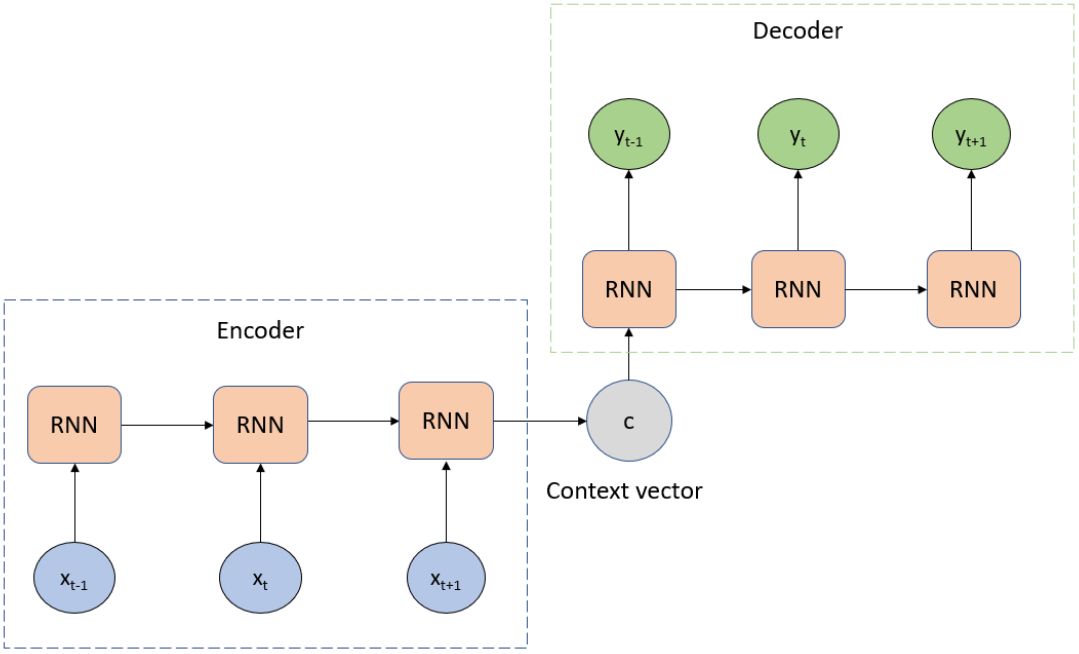

Figure 2.7: Example of an encoder-decoder architecture with RNN

2.2.2 Transformers

The attention mechanism proposed by Bahdanau in [BCB15] was implemented on an encoder-decoder architecture with bidirectional LSTM as an encoder for machine translation. It deals with the problem of fixed-length context vectors by enabling the decoder to focus on input using a sequence of context vectors. However, the benefit of the attention mechanism is limited to the decoder in the proposed architecture [BCB15]. As the architecture was based on RNN, which is sequential, parallelism could not be achieved during training. Transformers [VSP+] proposed by Vaswani overcomes the above challenges by implementing attention in the encoder and replacing RNNs. The proposed architecture used attention in both encoders and decoders (see Figure 2.8). Six encoders and decoders are stacked in the proposed architecture, and a residual connection is placed between the stacked layers. The encoder and decoder consist of a multi-head attention module and a feed-forward neural network. According to Vaswani [VSP+], attention is the process of mapping query and key-value pairs to output. The multi-head attention module performs the task of computing the attention using scaled dot product attention for a given input

by using query (Q), key (K), and value(V) vectors (see Figure 2.9). The mathematical representation of the scaled dot product attention from [VSP+] is mentioned in Equation 2.4. The dot product of Q and K is scaled using $\sqrt{d_k}$ to deal with the larger magnitude values, so it is termed as scaled dot product attention.

$$Attention(Q, K, V) = Softmax(\frac{(QK^T)}{\sqrt{d_k}})V \tag{2.4}$$

Where:

Q, K, V = query, key, and value vectors
d_k = dimension of key vector

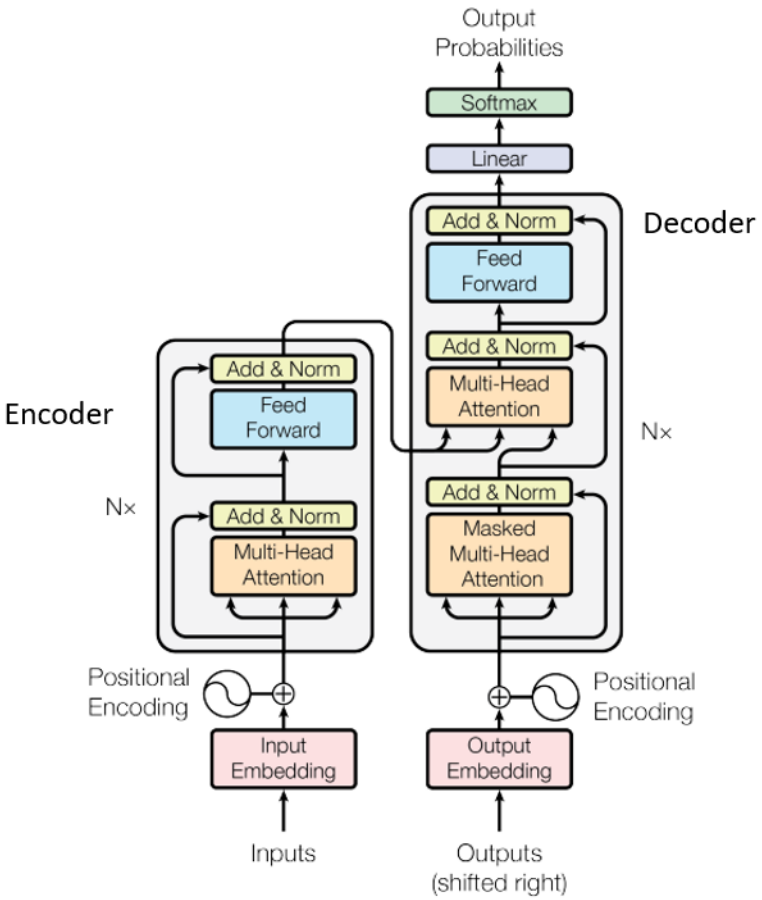

Figure 2.8: Transformer architecture from [VSP+]

The authors found it beneficial to perform the scaled dot-product attention multiple times by projecting the Q, K, and V vectors into different d_q, d_k, and d_v dimensions. The attention was applied on these vectors 'h' times as observed in Figure 2.9. The term 'h' also indicates the number of attention heads, indicating the number of times attention gets computed. Then the output after computing the attention for 'h' times is concatenated and projected as a final output. The representation of this

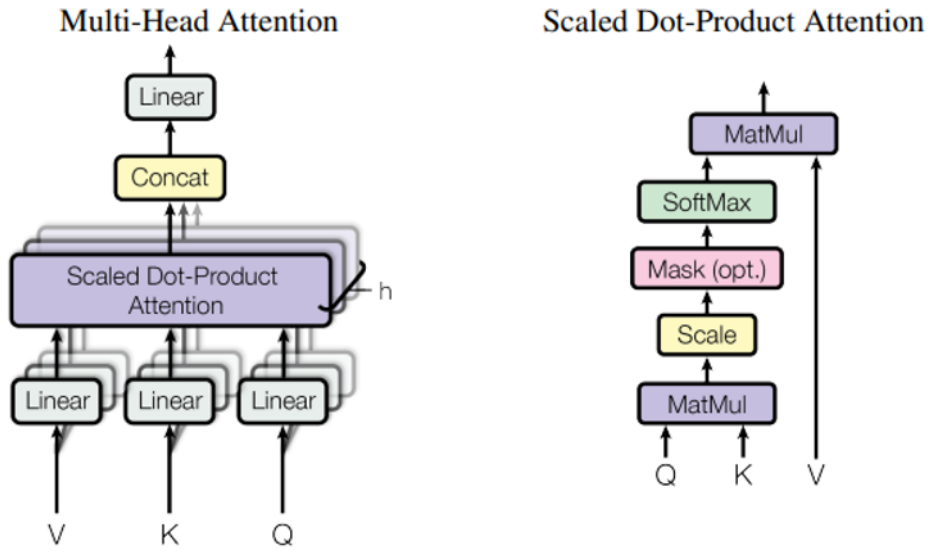

Figure 2.9: Attention mechanism in Transformers [VSP+]

process is indicated in the following equations Equation 2.5, and Equation 2.6 in [VSP+].

$$Multiheadattention(Q, K, V) = Concat(head_1, , .., head_h)W^o \tag{2.5}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{2.6}$$

Where W_i^Q,W_i^K, W_i^V, and W_i^O are parameter matrices of query, key, value, and output vectors.

The proposed architecture achieved better results than state-of-the-art methods for the machine translation task. In addition, the transformers were used for language representation in tasks related to Information retrieval, Question answering, and Text classification [WDS+]. The contextualized representation of the language as embeddings plays a crucial role in the tasks mentioned above. Since the representation provides vital information related to the meaning of the language or text, extracting these representations as embeddings using pre-trained language models became popular in NLP [HR]. In deep learning, pre-training refers to initially training a model on a task and then using its parameters to perform downstream tasks. ELMO [PNI+] and GPT [RN18] (Generative Pre-trained Transformer) are examples of such pre-trained language models. However, the authors of [DCLT] specify that the language models mentioned above are unidirectional. So they proposed BERT [DCLT] to overcome the limitation of unidirectionality by using masked language modeling (MLM) and Transformer architecture. Some other language models based on Transformer are SBERT (Sentence BERT)[RG], and T5[RSR+19]. In this thesis, we used BERT and SBERT for the implementation, so we discussed these architectures further.

2.2.3 BERT

BERT is the acronym for Bidirectional encoder representations from transformers. It provides a contextualized language representation using the attention mechanism proposed in [VSP+]. The BERT framework is based on pre-training and fine-tuning steps. The pre-training consists of tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). BERT tackles the problem of unidirectionality by considering the context in the left and right sides of the sequence in an MLM task. The main objective of the MLM task is to predict a missing word using the context in randomly masked sequences. It is specified in [DCLT] that 15% of the tokens in the input sequences are masked. These masked tokens are the targets that BERT predicted during this MLM pre-training task. However, Devlin et al. [DCLT] express the opinion that tasks like Question-Answering or Natural Language Inference (NLI) rely upon understanding the relationship between sentences, and language modeling alone cannot serve the purpose. So they incorporated a Next Sentence Prediction (NSP) task in the pre-training phase of BERT. In this task, given two sentences, A and B, BERT is trained to predict whether B is the following sentence of A. The training dataset in NSP was balanced for the scenario when B is the following sentence of A and when B is not the following sentence of A.

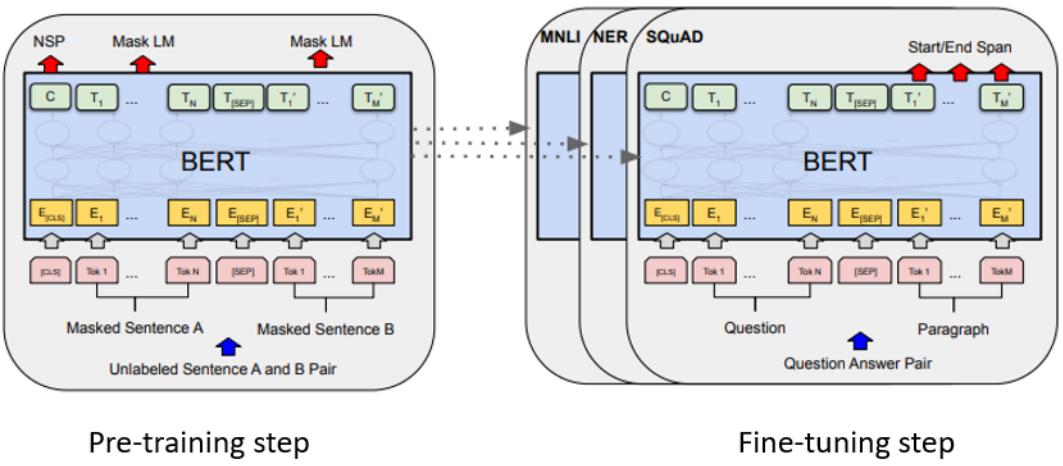

Figure 2.10: Overview of pre-training and fine-tuning steps of BERT [DCLT]

In fine-tuning step, the pre-trained BERT is trained on a downstream task using the task-specific dataset. In this step, BERT is initialized with pre-trained parameters from the pre-training step, and these parameters are updated based on the down-stream task. Figure 2.10 illustrates the pre-training and fine-tuning steps of BERT. It can be observed that for any down-stream task like Question-Answering (SQuAD) [RZLL] and Natural Language Inference (NLI) [WNB] the pre-trained parameters from BERT are used as the initial parameters and later fine-tuning is performed. For the MLM task, BERT takes one sentence or sequence as input and two sentences in the NSP task. In addition, for downstream tasks like Question-Answering (QA) and text classification, the input for BERT varies. So, BERT handles such ambiguity in the input through its input representation method. It combines multiple sentences

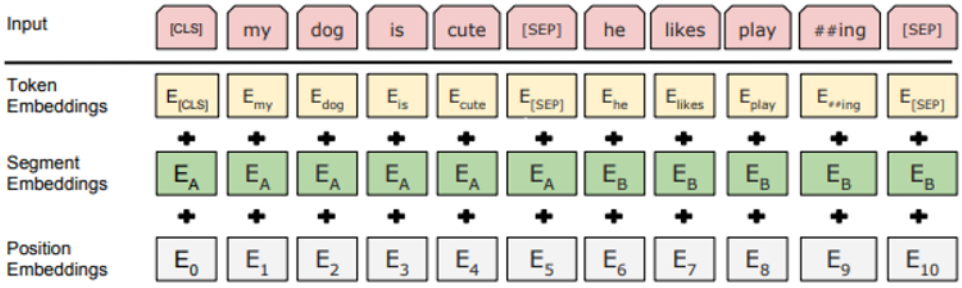

Figure 2.11: Input representation of BERT [DCLT]

into a single sequence by using a unique token '[SEP]' to distinguish multiple sentences (see Figure 2.11). Furthermore, it adds the position and segment embeddings to the token embeddings to represent each token in the input. These steps to handle ambiguity in input makes BERT suitable for different NLP tasks.

The authors of [DCLT] proposed two model sizes for BERT, namely BERT $_{\text{base}}$ and BERT $_{\text{large}}$. In this thesis, we have used 'deepset/gbert-base' a BERT $_{\text{base}}$ model by Deepset [CSM] available in HuggingFace [3] platform. We used the BERT model since the results compared with GPT-3 were satisfactory for the task of Occupation Coding. The 'deepset/gbert-base' model was pre-trained using German Wikipedia dump, OSCAR, OpenLegalData, and news articles datasets. Further, it was evaluated on GermEval18 [WS18], and GermEval14 [BBKP14] text classification datasets as downstream tasks. We used the German language BERT model since the dataset used in the thesis is in German. Even though BERT successfully set new state-of-the-art results in NLP, it faced limitations due to computational overhead and bad sentence embeddings in regression tasks like clustering and semantic similarity comparison. The authors of [RG] mention the above limitations in BERT and propose a modified version of BERT named Sentence-BERT (SBERT).

2.2.4 SBERT

SBERT is based upon BERT, Siamese, and triplet networks capable of deriving similar sentence embeddings for semantically similar sentences [RG]. The Siamese network [KZS15] based architecture of SBERT (see Figure 2.12) contains two identical neural networks that share weights. These networks work in parallel fashion, and their outputs are compared using cosine similarity. SBERT was evaluated on Semantic Textual Similarity (STS) and Argument Facet Similarity (AFS) datasets that compare the similarity of sentences. In this thesis, we have used SBERT for extracting embeddings in a similarity computation task based on the findings from [RG] that sentence embeddings from BERT are not suitable for calculating the similarity between sentences; we have used a German SBERT model from the HuggingFace [4] platform.

[3] https://huggingface.co/deepset/gbert-base
[4] https://huggingface.co/T-Systems-onsite/cross-en-de-roberta-sentence-transformer

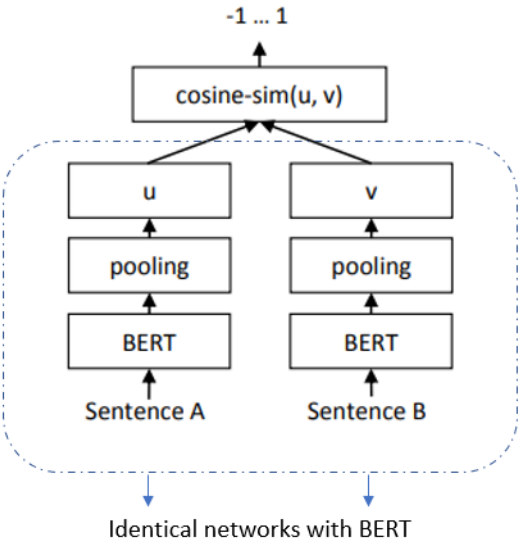

Figure 2.12: Architecture of SBERT [RG]

3. Related Work

This chapter discusses the different methods used to address the occupation coding task. Furthermore, as our proposed approach is based on imparting the domain knowledge of the KLDB classification scheme, we discuss the several methods in the literature which were used to incorporate external knowledge with BERT to solve different NLP tasks.

3.1 Occupation Coding - Methods

As mentioned earlier in Section 1.1, the occupation coding methods can be grouped into manual and (semi-)automatic methods. We discuss the state-of-the-art (semi-) automatic methods implemented to address the occupation coding task. In this section, we further differentiated the (semi-)automatic methods into rule-based, hybrid, and statistical methods with machine learning.

3.1.1 Rule-Based Methods

The rule-based methods in occupation coding assign an occupation ID to the preprocessed user responses by matching them against the rules provided by the domain experts in the 'if-then' format. In addition, these methods use occupation classification schemes like Table 2.1 and Table 2.4 as an index or dictionary for matching the user responses. Conrad et al. [Con97] in 1997 proposed an earlier implementation based on an expert system with pre-defined rules for occupation coding. These methods rely upon text-matching techniques and rules to classify user responses. Jürgen et al. [HZG03], and Hartmann et al. [HTS12] consists of rules to perform occupation coding using the ISCO-08 and KLDB 2010 schemes. However, the authors of [Sch14] mentioned that it is rare to achieve more than 50% of accuracy using rule-based methods. So, the usage of advanced text processing and text matching techniques became popular. Occupation coding systems like G-Code [1] by Statistics Canada and CASCOT [EPR14] by the Warwick Institute for Employment Research

[1] https://www150.statcan.gc.ca/n1/en/catalogue/10H0033

used the classification schemes combined with text processing techniques to classify occupations. CASCOT [EPR14] provides a confidence score after assigning an occupation ID. Another approach proposed in [BBA20] is an iterative algorithm that uses text search and filtering steps. It considers the user response as a query and searches the Canadian National Occupational Classification (NOC -2016) based on seven search strategies. It uses exact, partial, and weak text matching in the search steps to select an appropriate occupation ID for a query based on the job description and titles. If multiple occupation IDs match the description, it applies to a series of filtering steps that check keyword frequency in the job title or description from NOC and the query.

Additionally, some methods explored the use of linguistic relations between the user responses and descriptions of several occupation categories available in the classification schemes. These methods converted the responses and occupation descriptions into vectors and computed cosine similarity, then assigned the response to a category based on the highest similarity. Patil et al. [PP13] and Hacking et al. [HW12] utilized the linguistic relations between the texts for classification. However, the authors of [Sch14] mentioned that usage of statistical techniques with the Machine Learning algorithms exceeded the performance of the rule-based and text search methods.

3.1.2 Machine Learning - Methods

There has been considerable research on the automatic classification of texts into predetermined categories using Machine Learning (ML). The authors of [KJMH+19], [MP18], and [MKC+] provides an overview of the Machine Learning and Deep Learning (DL) methods for the text classification. [MKC+] also discusses how Deep Learning methods have outperformed the ML methods in the text classification tasks like sentiment analysis, news categorization, and NLI. As the occupation coding task deals with classifying the user responses into an occupation category by assigning an occupation ID, it can be treated as a text classification problem. However, considering it as a text classification problem involves additional challenges like less text and high dimensionality. In the first place, the user responses consist of keywords with minimal text and incorrect spellings. Furthermore, the responses have to be classified among several categories of occupations leading to the high dimensionality problem. Schierholz et al. [SS20] discuss the above challenges and address these issues by pooling data from multiple surveys to deal with the lack of text issue and also to ensure the presence of all the categories of occupations in the data.

In the context of Machine Learning, the methods implemented so far primarily represented the user responses as vectors using bag-of-words (binary presence, TF-IDF) from the human-annotated training data and used them as features to train the ML models. Ikudo et al. [ILSW20] used the bag-of-words to represent the occupation data and train an ML model using the random forest algorithm. Alexander et al. [Mea14] represented the occupation-related data of employees using n-grams and evaluated the performance of Multinomial Naive Bayes, SVM, and logistic regression algorithms for occupation coding, as part of analyzing the work-related illness and injuries in the United States. It is specified in [Mea14] that the SVM and logistic regression algorithms had better accuracy than Naive Bayes. Takahashi et

al. [TTTL14] proposed an occupation coding system based on bag-of-words and SVM combined with a rule base for the classification schemes like ISCO and SSM. In addition, the system provides a three-grade confidence score for the classified occupation ID. Gweon et al. [GSK+17] also utilized the bag-of-words technique and proposed an adapted nearest neighbor algorithm. It classifies the new user response by computing the cosine similarity between the vector representation of the new user response and the training data responses and assigns the occupation ID of the most similar response. The proposed algorithm also exceeded the performance of the SVMs with linear kernels on ALLBUS survey data.

Also, Schierholz et al. [SS20] provided a comprehensive survey of the Machine Learning methods for the occupation coding task. It compared the performances of seven methods on five survey datasets. Exact string matching, CASCOT [EPR14], Memory based reasoning [CMSW], Adapted nearest neighbor [GSK+17], Multinomial regression and Tree boosting (XGBoost) methods. It evaluated the methods' performances for the production and agreement rate metrics. It is mentioned in [SS20] that the Multinomial regression and Tree boosting (XGBoost) methods outperformed other methods in the occupation coding task.

Apart from the methods mentioned earlier, deep learning methods and pre-trained models were also utilized in tasks like skills classification and normalizing job titles for occupation data. Tran et al. [TVL21] proposed a multi-label classification technique for predicting job titles for a given job description. The proposed approach used a Bi-GRU-LSTM-CNN with pre-trained models like BERT and DistilBERT to predict a suitable job title for a job description. Decorte et al. [Dec21] proposed JobBERT, a method to normalize job titles. The process of job title normalization involves connecting free-form job titles to their most relevant standard titles. Also, Nigam et al. [NTTS20] proposed SkillBERT, a BERT-based model to extract embeddings as input features for grouping or classifying skills into competency groups. The competency groups are generally used to match the job skills and the applicants. During the literature survey, we observed that the information from classification schemes was used as an index or reference to generate rules for the task of occupation coding. However, we also identified that the information available in the classification schemes like KLDB and ISCO was not used along with the pre-trained models like BERT for the occupation coding task. Hence, we propose a novel approach to impart that information into BERT for occupation coding. In this process, we also explored the techniques used to incorporate external knowledge into BERT for specific NLP tasks.

3.2 Imparting external information into BERT

This section discusses methods that integrate external information with BERT to solve Classification, Machine Translation, and Natural Language Understanding (NLU) tasks. We categorized the external information into lexical, syntactic, and semantic categories to group the proposed methods.

3.2.1 Lexical Information

Lim et al. [LTM] mention that even though BERT can capture semantic information from the text, integrating corpus information along with BERT will be beneficial for

specific tasks. Lim et al. proposed an ensemble approach to use TF-IDF information and noun count as features of the corpus along with the sentence embeddings extracted from BERT to address the tasks of abuse detection (subtask A) and target detection (subtask B) in SemEval-2020 Task 12. Similarly, Prakash et al. [PTM] proposed an approach for stance detection task by using count-based features (TF-IDF) with RoBERTa. The proposed approach achieved state-of-the-art results for the stance detection task. Koufakou et al. [KPBP] proposed HurtBERT for detecting abusive language in texts on social media. It used an external lexicon HurtLex [2], a multilanguage lexicon of offensive words along with BERT. It integrated the embeddings and encodings of offensive words to detect abuse in texts. Yan et al. [YTM] mentioned that the pre-trained language models do not possess task-specific statistical or domain knowledge information. They hypothesize that such information helps in token classification tasks. The proposed approach uses count-based features and uses a CRF to improve the performance of the toxic span prediction task.

3.2.2 Semantic and Syntactic Information

Zhang et al. [ZWZ+20] specified that language models like ELMO, GPT, and BERT, which are often used for language representation, utilize the context-sensitive features of character or word embeddings. [ZWZ+20] points out that structured semantic information, which can provide rich semantic information, needs to be taken into account by these language models. Hence, they proposed SemBERT, a fine-tuned BERT that uses Semantic Role Labelling (SRL) to use the explicit contextual information in the text. Similarly, Sundararaman et al. [SSW+19] mentioned that the transformer architectures, which provide representations of tokens by considering the relationship among the other tokens in sequence, could also be benefited by explicitly providing syntactic information. So, [SSW+19] proposed to use Parts-of-speech (POS), case (categorical attribute - upper or lower), and subword position for each token as additional inputs to the transformer architecture so that the encoder pays attention to these syntactic features. Accordingly, Sundararaman et al. [SSW+19] modified the pre-trained BERT based to infuse syntax information to BERT and achieved state-of-the-art results in machine translation as well as several downstream tasks from GLUE benchmark. However, Bai et al. [BWC+21] mentioned leveraging the syntax trees to provide syntactic information to BERT rather than simple syntactic features like POS and subword position. Bai et al. [BWC+21] proposed Syntax-BERT to effectively ingest syntax trees at the pre-trained checkpoint of BERT and achieved consistent results over BERT and RoBERTa on NLU tasks. Table 3.1 provides an overview of the above-mentioned methods based on the type of external information integrated with BERT.

3.2.3 Additional pre-training

Apart from the semantic, syntactic, and lexical information, to improve the text classification performance and provide domain knowledge to BERT, Yu et al. [YTM] converted a multi-class classification task into a sentence pair classification task by

[2] https://github.com/valeriobasile/hurtlex

constructing auxiliary sentences from the corpus to incorporate task-specific knowledge into BERT. Yu et al. [YTM] specify that by constructing auxiliary sentences, the problem of limited supervised training data can be addressed, and the proposed method achieved state-of-the-art results in the case of multi-classification datasets. In addition, Yu et al. [YTM] proposed to use MLM and NSP to impart domain knowledge into BERT. Also, Brinkmann et al. [BB21] mentioned that the performance of pre-trained models on downstream tasks can be improved by adding additional pre-training steps using the domain-specific corpus. [BB21] addresses the task of hierarchical product classification to group product offers from online shops. It used Masked Language Modelling (MLM) as an additional pre-training step to provide domain knowledge to BERT regarding the hierarchy of products.

3.2.4 Summary

We discussed the rule-based and Machine Learning techniques proposed in the previous sections to solve the occupation coding task. However, we observed that the classification schemes were often used as an index or dictionary in the proposed techniques, and the pre-trained language models were not utilized for the occupation coding tasks. Hence, we propose to utilize the KLDB classification scheme for occupation coding. In this attempt, we explored the techniques used to incorporate external information with BERT to solve various NLP tasks. The Table 3.1 summarizes the different approaches proposed for imparting external information with BERT.

Author	Title	Lexical	Semantic	Syntactic	Pre-training
Lim et al.	Uob at semeval-2020task 12: Boosting bert with corpus level information.	✓			
Prakash et al.	Incorporating count-based features into pre-trained models for improved stance detection.	✓			
Koufakou et al.	Hurtbert: incorporating lexical features with bert for the detection of abusive language	✓			
Yan et al.	Uob at semeval-2021 task 5: Extending pre-trained language models to include task and domain-specific information for toxic span prediction.	✓			
Zhang et al.	Semantics-aware bert for language understanding.		✓		
Sundararaman et al.	Syntax-infused transformer and bert models for machine translation and natural language understanding.			✓	
Bai et al.	Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees			✓	
Yu et al.	Improving BERT-Based Text Classification With Auxiliary Sentence and Domain Knowledge				Auxiliary sentence, MLM, NSP
Brinkmann et al.	Improving Hierarchical Product Classification using Domain-specific Language Modelling				MLM

Table 3.1: Overview of proposed methods for integrating different types of textual information with BERT

4. Concept

This chapter discusses the approach implemented to integrate domain knowledge into BERT. We further present the details about the datasets used and discuss how we fine-tuned BERT on occupation-coding tasks using domain knowledge.

4.1 Datasets

In this thesis, we used two German language datasets for occupation coding: KLDB 2010 (KLDB data) published by the Bundesagentur für Arbeit (BA) and the DZHW occupation data provided by the DZHW institute.

4.1.1 DZHW Occupation data

It is anonymized data with KLDB IDs assigned to user responses, collected through occupation-related questions mentioned in Table 1.2. The KLDB IDs were assigned by human coders/experts following the KLDB 2010 [PM13] classification scheme since the respondents were from Germany. Figure 4.1 consists of examples taken from the DZHW occupation data. The dataset contains the following features: *"Job Title"*, *"Activity_1"* and *"Activity_2"* indicate activities for a corresponding *"Job Title"*. The DZHW data consists of 877 unique classes (KLDB IDs) and 56,206 rows with the features mentioned earlier.

KLDB ID	Job Title	Activity_1	Activity_2
83124	Sozialpädagogin	Case Management, interkulturelles Training, Netzwerkarbeit	beraten und begleiten v. Migranten
41264	wissenschaftlicher Mitarbeiter	selbständiges Forschen im Bereich der Virologie	virologische Forschung
31164	Bauingenieur, Ingenieur für	Wärmeschutz von Gebäuden	Planung von Bauvorhaben

Figure 4.1: Anonymized DZHW occupation dataset

In the DZHW data, we observed that the activities often contain less text and sometimes just keywords. This poses a challenge for the ML or DL models to be trained

for occupation coding. Since the DZHW data followed KLDB 2010 classification scheme, we assumed that providing the domain knowledge of the KLDB 2010 classification scheme to BERT would improve its classification performance on the DZHW occupation data.

4.1.2 KLDB Data

We address this dataset as domain data since it contains details about the 5-level hierarchy of occupation groups. The information regarding the occupation group can be observed in Figure 4.2. The information provided in the KLDB columns is as follows:

- Ebene: indicates the hierarchy level
- Titel: contains a title for an occupation group
- Allgemeine Bemerkungen: contains description of the occupation group
- Einschlüsse: contains information about the activities and skills of an occupation group
- Umfasst ferner: contains the list of occupation groups and job titles that are included under an occupation group/title
- Ausschlüsse: contains the list of occupation groups and job titles that are excluded for an occupation group/title

From this dataset, we have used the occupation groups that belong to the fifth level since it contains information about the previous four levels. This dataset consists of 1286 rows for the fifth hierarchy level, which is further used to create custom datasets for fine-tuning tasks. In addition, we observed that the number of tokens in the description and activities for both DZHW and KLDB data didn't exceed 512, which is the maximum input length for BERT.

KLDB ID	Ebene	Titel	Allgemeine Bemerkungen	Einschlüsse	Umfasst ferner	Ausschlüsse
1	1	Land-, Forst- und Tierwirtschaft und Gartenbau			Die Systematikposition umfasst folgende Unterpositionen: 11 Land-, Forst- und	
11	2	Land-, Tier- und Forstwirtschaftsberufe	Inhalt: Diese Systematikposition umfasst die Berufe in der Land-, Tier-, Pferde- und Fischwirtschaft, Tierpflege, Weinbau sowie Forst- und Jagdwirtschaft und Landschaftspflege.		Die Systematikposition umfasst folgende Unterpositionen: 111 Landwirtschaft 112 Tierwirtschaft 113 Pferdewirtschaft 114	
111	3	Landwirtschaft	Inhalt: Angehörige dieser Berufe übernehmen Aufgaben in der Landwirtschaft, in der Landtechnik oder im	Aufgaben, Tätigkeiten, Kenntnisse und	Die Systematikposition umfasst folgende Unterpositionen: 1110	Nicht einzubeziehende Positionen: 112 Tierwirtschaft 114 Fischwirtschaft 115
1110	4	Berufe in der Landwirtschaft (ohne Spezialisierung)	Inhalt: Angehörige dieser Berufe sind in der Herstellung landwirtschaftlicher Produkte tätig. Sie arbeiten in der Pflanzen-	Aufgaben, Tätigkeiten, Kenntnisse und	Die Systematikposition umfasst folgende Unterpositionen:	Nicht einzubeziehende Positionen: 1121Berufe in der Nutztierhaltung (außer
11101	5	Berufe in der Landwirtschaft (ohne Spezialisierung) - Helfer-	Inhalt: Diese Systematikposition umfasst alle Berufe in der Landwirtschaft, deren Tätigkeiten in der Regel keine speziellen Fachkenntnisse erfordern. Angehörige	Aufgaben, Tätigkeiten, Kenntnisse und Fertigkeiten,	Zugeordnete Berufe (Beispiele): Landwirtschaftliche/r Helfer/in Erntehelfer/in	Nicht einzubeziehende Berufe: Tierwirtschaftshelfer/in (11211) Fischereihelfer/in (11401) Forstwirtschaftshelfer/in

Figure 4.2: KLDB dataset with description and activity information for occupations

4.2 Implementation

In this section, we discuss how domain knowledge was provided to BERT using two dimensions: Training and Dataset dimensions. In the training dimension, we discuss the two approaches and the tasks involved in each approach to provide domain knowledge and perform occupation coding. Also, we discuss the steps performed in the Dataset dimension.

4.2.1 Training dimension

To integrate the KLDB classification scheme information into BERT, we propose to use additional pre-training steps with a series of classification tasks. Hence, we propose two approaches, namely, Approach 1 & 2, in which we fine-tune BERT on text binary classification and text classification tasks for the KLDB data. These approaches are based on Yu et al. [YSL19] & Brinkmann et al. [BB21] which suggest to use NSP and auxiliary sentence classification tasks to integrate domain knowledge into BERT. To utilize the domain information provided in KLDB data for occupation coding, we decided first to fine-tune BERT on binary classification and occupation coding on the domain data and later utilize the fine-tuned BERT for occupation coding on DZHW occupation data (see Figure 4.3). The motivation behind these approaches was also to tackle the lack of text in DZHW data by imparting the KLDB classification scheme knowledge to BERT.

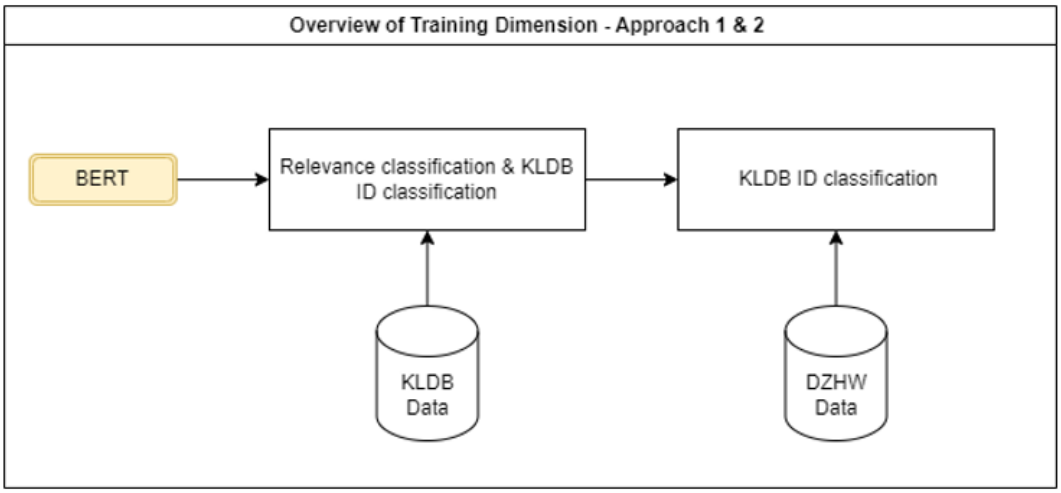

Figure 4.3: Overview of the tasks involved in training dimension

Approach 1:

In this approach, we fine-tuned BERT on a text classification task (KLDB ID Classification) in which we provided the job activity, title, and description as input to BERT and classified it into a corresponding KLDB occupation group using the KLDB ID. We considered this fine-tuning task as a method to integrate the domain

knowledge into BERT. We created a custom dataset from KLDB data to perform this task since it only had 1286 rows of data for the five-digit KLDB IDs. Hence, we created a custom dataset from the occupation activities and description information. As the KLDB data contains activities and descriptions for occupations, we assumed that if BERT can perform well in predicting the KLDB ID on a given input of job title and activity from domain data, it would simultaneously increase the performance on the DZHW occupation data. The pipeline in Figure 4.4 depicts the steps involved in this approach.

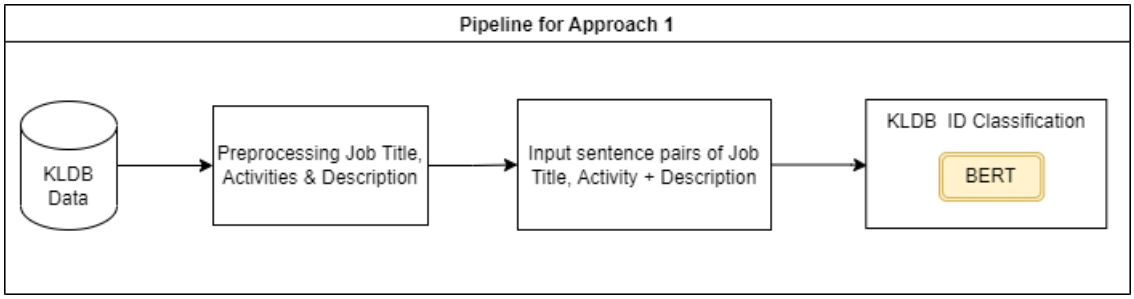

Figure 4.4: Pipeline 1 - BERT is fine-tuned to perform occupation coding on a custom dataset created from KLDB data.

Custom Dataset - Approach 1

To prepare a custom dataset, we considered Allgemeine Bemerkungen (Description) and Einschlüsse (Activity) columns apart from KLDB ID and Titel (Title) in KLDB Data because the activities and Job Titles of user responses in the DZHW occupation data contained semantically similar words. In the process of custom data creation, the activities featured in the Einschlüsse column for a corresponding Job Title (see Figure 4.5) were split into a list of sentences so that more instances of data could be generated (see Figure 4.6). Then we performed basic preprocessing steps like removing special characters, replacing abbreviations in the activities, and rearranging the titles. Since some titles were of the form *"Komiker/innen und Kabarettisten/Kabarettistinnen"* after the preprocessing step, they were converted into *"Komiker oder Komikerinnen und Kabarettisten oder Kabarettistinnen"*. After performing basic preprocessing steps, we created a custom dataset with 8,907 instances from the 1286 instances. In addition, we added the 4200 job titles extracted from Ausschlüsse but they lacked the activity and description information.

KLDB ID	Title	Activity
11101	Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder AnlerntÃ¤tigkeiten	Gras, Heu oder Stroh zusammenrechen und laden . Pflanzen mit Hilfe von Handwerkzeugen bewässern, ausdünnen und jäten . Hilfsarbeiten bei der Ernte durchführen . Obst, Nüsse, Gemüse und andere Früchte pflücken oder aufsammeln . Tiere füttern, tränken und Tierunterkünfte reinigen.Reinigungsarbeiten an landwirtschaftlichen Einrichtungen und Maschinen durchführen.kleinere Reparaturen an Tierunterkünften und landwirtschaftlichen Einrichtungen durchführen . Hilfsmittel, Erzeugnisse und sonstige Materialien, z.B. Futtermittel, tierische und pflanzliche Produkte be- und entladen

Figure 4.5: Activities before splitting for KLDB ID - 11101

KLDB ID	Title	Activity
11101	Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten	Gras, Heu oder Stroh zusammenrechen und laden.
11101	Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten	Pflanzen mit Hilfe von Handwerkzeugen bewässern, ausdünnen und jäten.
11101	Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten	Hilfsarbeiten bei der Ernte durchführen.
11101	Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten	Obst, Nüsse, Gemüse und andere Früchte pflücken oder aufsammeln.
11101	Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten	Tiere füttern, tränken und Tierunterkünfte reinigen.Reinigungsarbeiten an landwirtschaftlichen Einrichtungen und Maschinen durchführen.
11101	Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten	kleinere Reparaturen an Tierunterkünften und landwirtschaftlichen Einrichtungen durchführen.
11101	Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten	Hilfsmittel, Erzeugnisse und sonstige Materialien, z.B. Futtermittel, tierische und pflanzliche Produkte be- und entladen

Figure 4.6: Activities after splitting for KLDB ID - 11101

We considered KLDB ID as the target variable from the custom data and assumed this as an occupation coding task on domain data (KLDB ID Classification) and fine-tuned BERT. During KLDB ID Classification, a new classification head was added on top of BERT to perform the classification to match the number of unique KLDB IDs in the domain data. Also, during KLDB ID Classification, the pre-trained and classification head weights of BERT get updated. It was also ensured that the train, test, and validation splits contained the examples for all the KLDB IDs. After the KLDB ID Classification, we assume that BERT contains the domain knowledge of KLDB, so we only discard its classification head and replace it with a new head to match the number of unique KLDB IDs in the DZHW occupation data and fine-tune it again on the DZHW occupation data. We discuss this approach's hyperparameter configuration and results in the experiments section.

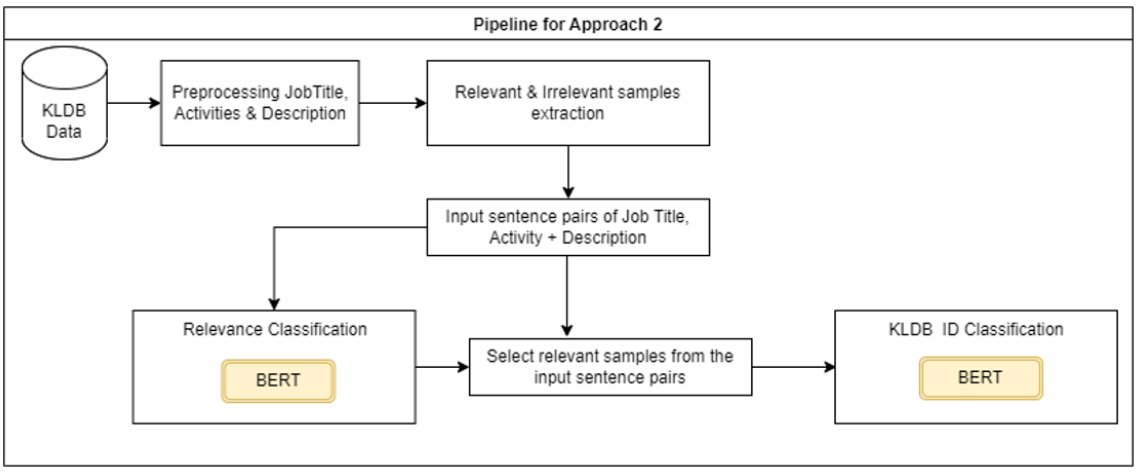

Figure 4.7: Pipeline 2 - BERT fine-tuned on an additional sentence pair classification and occupation coding tasks on domain data.

Approach 2:

In this approach, we use an auxiliary sentence pair classification task as an ad-

ditional step before performing the fine-tuning step on domain data as depicted in the Figure 4.7. In the auxiliary task, BERT is fine-tuned to classify whether an activity, description, and job title are relevant or irrelevant to each other. We address this as a relevance classification task in Figure 4.7. This task was inspired by the NSP task mentioned in Section 2.2.3. Since several occupation groups or titles in KLDB data have semantically similar words in the activities and description, we assumed that if BERT can classify whether an activity and title are relevant to each other would further help improve its performance during the occupation coding on domain and DZHW data.

KLDB ID	Relevance	Activity	Title
11101	1	Diese Systematikposition umfasst alle Berufe in der Landwirtschaft, deren Tätigkeiten in der Regel keine speziellen Fachkenntnisse erfordern. Angehörige dieser Berufe führen in landwirtschaftlichen Betrieben nach Anweisung einfachere oder zuarbeitende Tätigkeiten aus. Je nach Einsatzgebiet bearbeiten sie die Erde, versorgen Nutztiere und übernehmen Pflanz- und Erntearbeiten.Hilfsmittel, Erzeugnisse und sonstige Materialien, zum Beispiel Futtermittel, tierische und pflanzliche Produkte be und entladen.	Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten
11101	0	Diese Systematikposition umfasst alle Berufe in der Nutztierhaltung, deren Tätigkeiten fundierte fachliche Kenntnisse und Fertigkeiten erfordern. Angehörige dieser Berufe züchten, halten und versorgen Nutztiere, insbesondere Rinder, Schafe und Schweine, um die Tiere bzw. die entsprechenden Tierprodukte zu vermarkten.Weideland, Futter und Wasserbestände bereitstellen und überwachen.	Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten

Figure 4.8: Examples of relevant and irrevalant instances for the occupation - ”Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten”

Custom Dataset - Approach 2

We have created a custom dataset based on the nearest neighbor approach from the KLDB domain data to perform the binary classification task. We considered the provided activity information in domain data as relevant samples in the custom dataset creation for occupations. For example, the activity information provided in Figure 4.5 is positive/relevant for the title *”Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten”*. To create an irrelevant sample, we had a couple of alternatives: Firstly, based on the nearest KLDB ID and a similarity-based method. In the KLDB ID approach, we consider the activities of the nearest KLDB ID as irrelevant samples. For example, for the KLDB ID 11102, the irrelevant samples will be the activities of KLDB IDs - 11101 and 11103. So, the nearest KLDB IDs are determined by adding and subtracting one from a KLDB ID and verifying whether that KLDB ID exists in the classification scheme. However, we pursued the idea of using the similarity-based approach because we assumed that training BERT to classify semantically similar activities for a KLDB ID would further improve its performance in occupation coding tasks on domain and DZHW data.

Algorithm 4.1: Relevant and irrelevant sample extractor

```
   Result: Relevant & irrelevant instances for each occupation title
 1 occupation_list: list of occupations in KLDB;
 2 activity_list: list of activities for each occupation in KLDB;
 3 activity_embeddings: Sentence BERT embeddings for activities of each
     occupation in KLDB;
 4 similarity_list: list of activity similarity values for each occupation;
 5 rel_irr_samples: relevant and irrelevant samples for each occupation;
 6 for occupation in occupation_list do
 7 |   activity_embeddings[occupation] = SBERT(activity_list[occupation])
 8 end
 9 for current_occupation in occupation_list do
10 |   for other_occupation in occupation_list do
11 |   |   if current_occupation is not other_occupation then
12 |   |   |   similarity_list [current_occupation] = similarity
   |   |   |     (activity_embeddings [current_occupation, other_occupation]);
13 |   |   end
14 |   |   else
15 |   |   |   rel_irr_samples[current_occupation]['rel'] =
   |   |   |     activity_list[current_occupation]
16 |   |   end
17 |   end
18 |   most_similar_occupation = argmax(similarity_list
   |     [current_occupation])
19 |   rel_irr_samples [current_occupation]['irrel'] =
   |     activity_list[most_similar_occupation]
20 end
21 return rel_irr_samples;
```

In the similarity-based approach, we utilize the text embeddings of the activities for occupations provided by the KLDB scheme. For example, for the KLDB ID 11101 *"Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten"*, to determine an irrelevant sample, we take the embeddings of the activities for 11101 and all the other KLDB IDs. Then, we determined the most similar activity based on cosine similarity and considered it an irrelevant sample. The Figure 4.8 depicts an example of relevant and irrelevant instances for the occupation with KLDB ID 11101. The relevance column indicates whether the text description of the activity is relevant to the occupation *"Berufe in der Landwirtschaft ohne Spezialisierung Helfer oder Anlerntätigkeiten"*. We further explain the relevant and irrelevant sample extraction in Figure 4.7 based on the algorithm 4.1.

From lines 6 to 8, it can be observed that SBERT (sentence BERT) was used for generating the embeddings of the activities since it is mentioned in [RG] that sentence embeddings generated from BERT are not suitable for similarity computation tasks. As the positive/negative sample extraction is based on computing cosine similarity for the embeddings, we decided to use SBERT. From lines 9 to 19, for each occupation in the domain data, we compute cosine similarity for the activities (lines 10-13) and consider the current occupation's activity as a positive sample (line 15)

and only consider the topmost similar activity (line 18-19) as a negative sample. Since there can be a lot of irrelevant samples for an activity, we decided to consider only the topmost or first activity in the similarity list as an irrelevant sample so that BERT can differentiate between the closely related activities of different job titles and also avoid the class imbalance scenario in case of relevant and irrelevant samples.

The custom dataset consisted of 18,251 instances for the binary classification task. As observed in the Figure 4.7, we first fine-tune BERT on the binary classification task (Relevance classification). Then we chose only the relevant/positive examples from the created custom dataset and fine-tuned the BERT model from the binary classification task to perform occupation coding on the domain data. Further, we use the BERT model fine-tuned on domain data to perform occupation coding on DZHW occupation data. The difference between approaches 1 and 2 is using the binary classification task (Relevance classification) as an additional step before fine-tuning BERT on domain data.

4.2.2 Dataset dimension

In the dataset dimension, we address the research question RQ-3 by augmenting the user responses in the DZHW occupation data with the domain data to provide more textual information as input to the BERT model during the training phase. We append the DZHW data and the positive/relevant samples from the custom dataset in approach 2 and fine-tuned BERT on the augmented data to perform occupation coding. We also ensured that the domain data existed only in the training and validation datasets but not in the test dataset to judge whether adding the domain data to user responses in DZHW occupation data improved BERT's classification performance.

To summarize, we created two BERT models from the Training dimension, i.e., from Approaches 1 and 2, and one BERT model from the Dataset dimension. The domain knowledge of the KLDB classification scheme was provided through a series of classification tasks in the Training dimension, whereas in the Dataset dimension, we provided the domain knowledge to BERT along with DZHW occupation data as an input during the training phase. To judge whether integrating the domain knowledge into BERT to perform occupation coding on DZHW occupation data improved the classification performance, we have created a baseline BERT model. This baseline model was fine-tuned to perform occupation coding on the DZHW occupation data without any information regarding the KLDB classification scheme.

5. Experiments

In this chapter, we discuss the experimental setup and the experiments performed for the KLDB ID classification and relevance classification tasks on the domain data and later about the hyperparameter tuning configurations and evaluation of the performance of BERT for the occupation coding tasks involved in the proposed approaches.

5.1 Experimental setup

We implemented data understanding and data preparation, modeling, and evaluation steps using the Python programming language for the proposed approach. In the data understanding and data preparation steps, the KLDB and DZHW occupation datasets were analyzed and preprocessed using the Pandas[1] and Spacy[2] libraries. In the modeling step, we performed hyperparameter tuning and fine-tuning steps on the BERT models using PyTorch [PGM+19], Transformers [WDS+] by Hugging Face, and Tune: scalable hyperparameter tuning framework [LLN+18] from Ray[3] libraries. Furthermore, we used the Weights & Biases [Bie20] tool to track the experiments. During the hyperparameter tuning step, the best configuration was chosen based on the performance of BERT on the validation dataset. After finding the best hyperparameter configuration, we fine-tuned the BERT model following the steps proposed for the Training and Dataset dimension and baseline.

After the fine-tuning step, we performed a stratified k-fold cross-validation (k=5) step for the DZHW occupation data to evaluate the performance of models based on accuracy. In the stratified cross-validation step, the DZHW occupation data was split into five folds of training and test dataset splits. The training and test splits for the baseline and training dimension consisted of 44,920 and 11286 instances. As we augment the domain data with DZHW data for dataset dimension the training and test splits contain 58,024 and 11286 instances. For each fold, a new BERT

[1] https://pandas.pydata.org/docs/
[2] https://spacy.io/
[3] https://docs.ray.io/en/latest/index.html

model was fine-tuned on training data, and the performance on the test dataset was recorded. This step aims to determine how the performance of BERT varied for different training and test datasets. As the DZHW data was imbalanced for the 877 unique KLDB IDs, we ensured that examples from each class were present in training and test data splits. To perform the experiments mentioned in the modeling and evaluation steps, we used the Tesla V100-SXM2-32GB GPU.

5.2 Initial experiments

This section discusses the initial experiments performed on the KLDB and DZHW occupation data and the reason for choosing the grid search method to perform the hyperparameter tuning step.

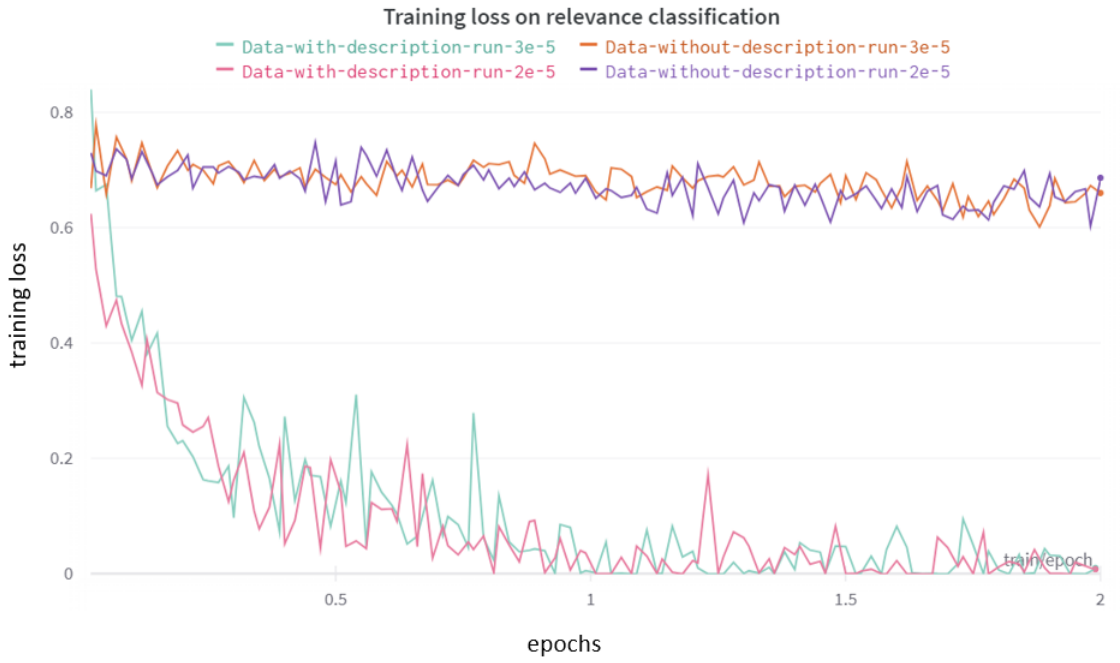

Figure 5.1: Training loss for relevance classification task reduced after adding the description to activities for learning rates $2e^{-5}, 3e^{-5}$

As mentioned earlier, the KLDB domain data comprised information about the occupation group's description and activities. We initially considered only the activity information from the domain data (Section 4.1.2) for the relevance classification task mentioned in the Approach 2. We observed that for a choice of the learning rates $2e^{-5}, 3e^{-5}$, and batch size 16, the training loss of the BERT model for the relevance classification on the domain data remained stagnant and didn't decrease as the training epochs were finished. This phenomenon can be observed from the Figure 5.1 indicated by 'Data-without-description'. After adding the occupation description to the activities, the training loss for the relevance classification started to decrease as indicated by 'Data-with-description' in Figure 5.1. Hence, we decided to utilize the description of occupations from the domain data in the proposed approaches. Furthermore, we observed that the BERT model required more than three epochs

to train on the domain data and the DZHW occupation data for the KLDB ID classification tasks.

In addition, we have experimented with two methods for the hyperparameter tuning step: Population-based training (PBT) [4] and Grid search. For 20 combinations of the batch_size, weight decay, learning rate, and epochs, we observed that the PBT method executed the hyperparameter tuning of BERT for the occupation coding task on DZHW data for six days. In contrast, the grid search method for 24 combinations of hyperparameters executed the hyperparameter tuning step for two days. Since PBT trains a series of models by mutating the hyperparameters and utilizing the parameters of other hyperparameter configurations, it took a lot of computing power and time. Table A.1 depicts the results of the baseline BERT model for hyperparameter tuning on the KLDB ID classification task for the DZHW occupation data. In addition, PBT provides a trained model and training schedules instead of the best hyperparameter configuration. So, we decided to use the grid search method instead of PBT for the hyperparameter tuning. Furthermore, we discuss the choice of hyperparameters and their values used for the hyperparameter tuning step.

Hyperparameter	**Values**
batch_size	[16, 32]
weight decay	(0, 0.3)
learning rate	$[2e^{-5}, 3e^{-5}, 4e^{-5}, 5e^{-5}]$
epochs	[5, 7, 9]

Table 5.1: Grid search hyperparameters for occupation coding tasks

To perform hyperparameter tuning for the classification tasks described in the proposed approaches, we have used the batch size, epochs, weight decay, and learning rate as hyperparameters. In the case of learning rate and batch size values, we have selected the suggested values mentioned in [DCLT] and used the suggested weight decay values by Tune [5] (see Table 5.1). We used the learning rate values suggested by [DCLT] after comparison of training loss and validation accuracy for other learning rates $1e^{-6}, 1e^{-7}$, and $5e^{-7}$ with one of the learning rate value $5e^{-5}$ from [DCLT] on KLDB ID classification task for DZHW data. It can be observed from the Figure 5.2 and Figure 5.3 that learning rate from $5e^{-5}$ tends to reduce the training loss and attain a better validation accuracy. In addition, we used the Adam optimizer and default dropout 0.1 value mentioned in [DCLT]. As discussed earlier, we have used the Tune framework from Ray to perform hyperparameter tuning in PyTorch on the 'gbert-base' model from the Hugging Face platform. The Tune framework creates different combinations of hyperparameters called trails using the grid search method, trains a BERT model on each trail, and suggests the best configuration of hyperparameters based on performance metrics like training loss or accuracy on a validation dataset, etc. We have considered accuracy as an evaluation or performance metric since we wanted to improve the accuracy of BERT for the occupation

[4] https://www.deepmind.com/blog/population-based-training-of-neural-networks

[5] https://medium.com/distributed-computing-with-ray/hyperparameter-optimization-for-transformers-a-guide-c4e32c6c989b

coding task. Moreover, as the KLDB ID classification is a multi-class classification problem, the micro and weighted average of f1-scores were similar to the accuracy. In addition, we have set a random seed 42 for all the hyperparameter tuning and cross-validation steps.

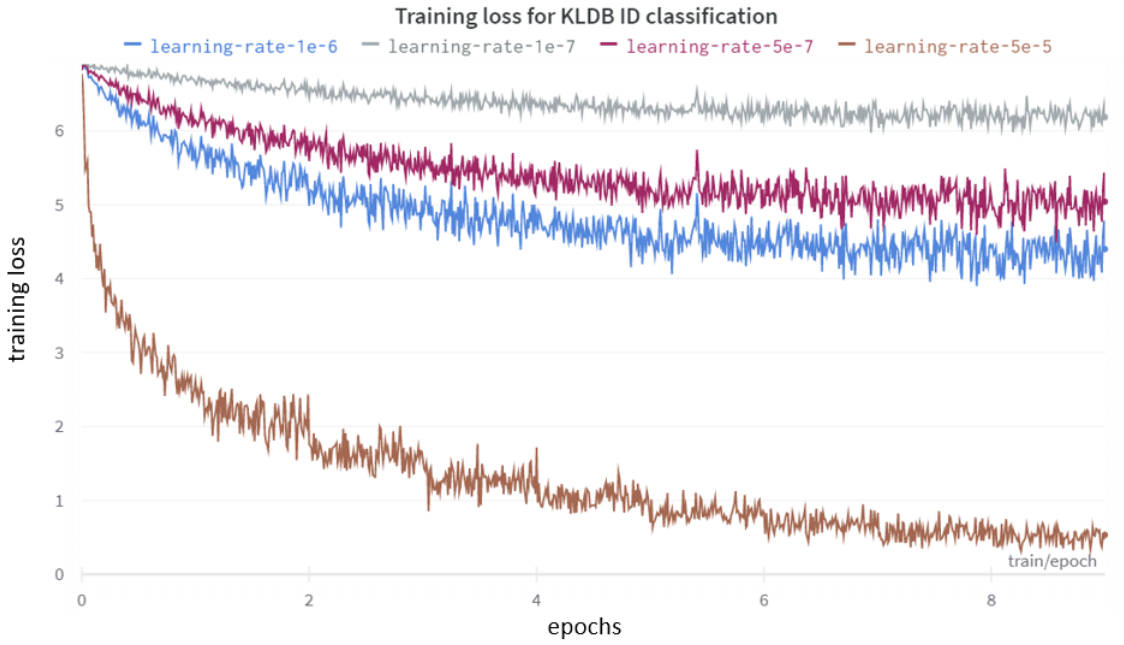

Figure 5.2: Training loss of BERT for KLDB ID classification task on DZHW data for learning rates $5e^{-5}, 1e^{-6}, 1e^{-7}$, and $5e^{-7}$

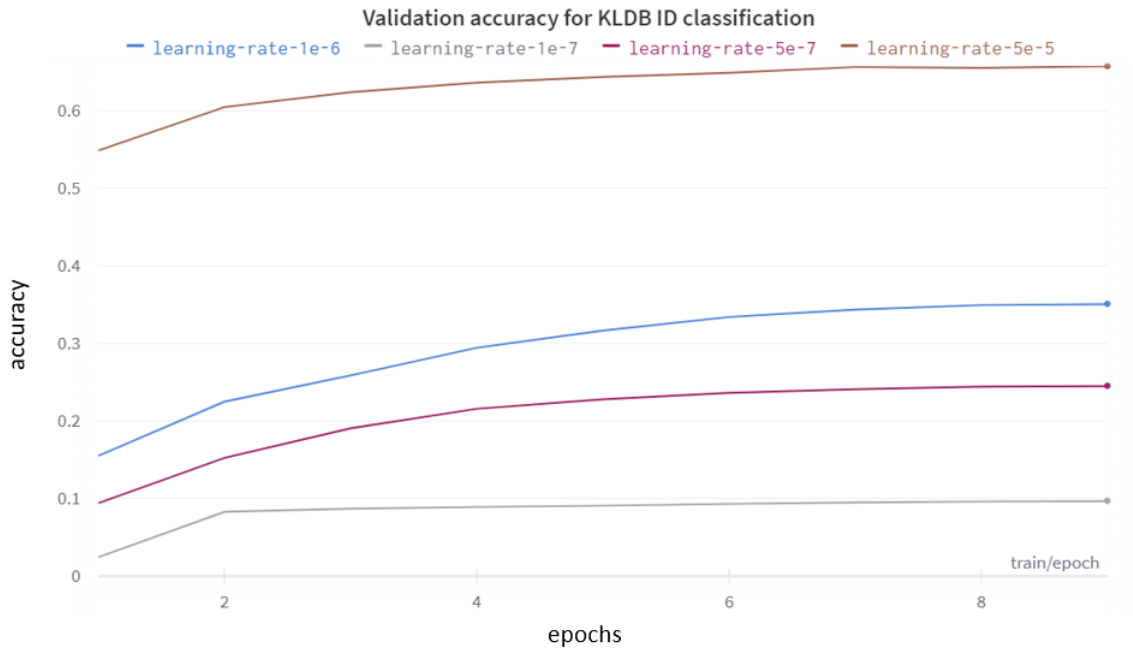

Figure 5.3: Validation accuracy of BERT for KLDB ID classification task on DZHW data for learning rates $5e^{-5}, 1e^{-6}, 1e^{-7}$, and $5e^{-7}$

5.3 Fine-Tuning the Models

This section presents the results of the hyperparameter tuning, fine-tuning and cross-validation steps for the BERT models from Training and Dataset dimensions and the baseline.

5.3.1 Approach 1

As discussed earlier in Section 4.2.1, approach 1 involves fine-tuning BERT on the domain data for the KLDB ID classification task; we have done the hyperparameter tuning on 'gbert-base' model with the parameter space from Table 5.1. We have performed the hyperparameter tuning with 24 trials generated from grid search and trained a BERT model for each run on the domain data training dataset. We have used validation accuracy as a metric to choose the best hyperparameter configuration. The Table 5.2 consists of the results for the hyperparameter tuning on the custom dataset generated from the KLDB domain data. The configuration of trial 15 performed well in comparison with the other trials in terms of validation accuracy. So using the best configuration suggested from the hyperparameter tuning step, we have trained a 'gbert-base' on the custom domain data and checked its performance on a test dataset from the custom dataset. The model achieved similar performance on the test dataset with 0.9736 (97.36 %) accuracy for the KLDB ID classification task.

After training BERT on the custom dataset, we assumed that the domain knowledge from KLDB data was imparted into BERT and fine-tuned it on the KLDB ID classification for anonymized DZHW data. We performed the hyperparameter tuning step on the fine-tuned BERT to choose the best parameters for the DZHW occupation data. The results of the hyperparameter tuning step for DZHW data can be observed in the Table 5.3 and trial 7 achieved a higher validation accuracy of 65.37 %. Furthermore, we have selected the hyperparameter configuration of the trial 7 and performed stratified k-fold cross-validation with k=5 on the DZHW occupation data to train the BERT with KLDB scheme information on different training data and evaluate the performance on different splits of test data. We have chosen stratified k-fold-cross-validation with to ensure that the training and test datasets consisted of instances belonging to all the KLDB IDs. The mean accuracy of the model on the test datasets from the stratified k-fold-cross-validation step was 66.67% with a standard deviation of 0.11.

5.3.2 Approach 2

As mentioned earlier in Section 4.2.1, Approach 2 involves relevance classification, an auxiliary sentence classification task as an additional step before fine-tuning BERT on the domain data for the KLDB ID classification task. We have first fine-tuned the 'gbert-base' model on the custom dataset created from relevant and irrelevant sample extractor for relevance classification, a binary classification task with the following parameters from hyperparameter tuning step: 3 epochs, 16 input batch size, and $3e^{-5}$ learning rate. The BERT model achieved an accuracy of 95.4% on validation and 95.12% on the test dataset for the relevance classification task. Further, we have fine-tuned this model on the KLDB ID classification task for the relevant

Trial	w_decay	lr	train_bs	epochs	val_acc	val_loss
0	0.0550304	$5e^{-5}$	16	5	0.00210	7.16355
1	0.0468056	$4e^{-5}$	16	5	0.87719	3.75629
2	0.137775	$3e^{-5}$	16	9	0.96491	2.10507
3	0.00617535	$2e^{-5}$	32	9	0.89754	4.62434
4	0.0637017	$5e^{-5}$	32	7	0.94877	2.8497
5	0.185244	$2e^{-5}$	32	5	0.57543	5.8279
6	0.129584	$3e^{-5}$	32	5	0.78666	5.09208
7	0.119958	$5e^{-5}$	16	9	0.96350	2.37206
8	0.136821	$4e^{-5}$	32	9	0.93473	2.77608
9	0.114739	$5e^{-5}$	32	9	0.96350	1.91287
10	0.0139351	$4e^{-5}$	32	5	0.77333	4.63553
11	0.13515	$2e^{-5}$	16	5	0.00140	7.164
12	0.28969	$2e^{-5}$	32	5	0.62386	5.78572
13	0.00478988	$2e^{-5}$	32	5	0.64070	5.77945
14	0.204979	$5e^{-5}$	32	9	**0.96701** ***	1.92.836
15	0.0520094	$3e^{-5}$	16	9	**0.97122** *	2.06761
16	0.198757	$2e^{-5}$	32	7	0.82315	5.17846
17	0.0623825	$5e^{-5}$	32	7	0.93824	2.794
18	0.290875	$2e^{-5}$	32	5	0.63859	5.77572
19	0.118545	$5e^{-5}$	32	7	0.94947	2.75407
20	0.276562	$5e^{-5}$	32	9	**0.96912** **	1.90421
21	0.15625	$2e^{-5}$	32	9	0.90105	4.59276
22	0.116603	$2e^{-5}$	32	5	0.63017	5.7915
23	0.176025	$3e^{-5}$	16	5	0.88140	4.05215

Table 5.2: Hyperparameter tuning results for KLDB ID classification on custom domain data. w_decay - weight decay, lr - learning rate, train_bs - batch size, val_acc - validation accuracy, val_loss - validation loss

Trial	w_decay	lr	train_bs	epochs	val_acc	val_loss
0	0.0550304	$5e^{-5}$	16	5	0.64083	1.82234
1	0.0468056	$4e^{-5}$	16	5	0.63611	1.88301
2	0.137775	$3e^{-5}$	16	9	0.64596	1.85746
3	0.00617535	$2e^{-5}$	32	9	0.62240	2.02624
4	0.0637017	$5e^{-5}$	32	7	0.64402	1.82448
5	0.185244	$2e^{-5}$	32	5	0.57721	2.37128
6	0.129584	$3e^{-5}$	32	5	0.60674	2.12088
7	0.119958	$5e^{-5}$	16	9	**0.65379 ***	1.83981
8	0.136821	$4e^{-5}$	32	9	0.64663	1.84414
9	0.114739	$5e^{-5}$	32	9	0.64840	1.81769
10	0.0139351	$4e^{-5}$	32	5	0.62383	1.9835
11	0.13515	$2e^{-5}$	16	5	0.60405	2.16116
12	0.28969	$2e^{-5}$	32	5	0.57746	2.37385
13	0.00478988	$2e^{-5}$	32	5	0.57754	2.36963
14	0.204979	$5e^{-5}$	32	9	**0.65076 *****	1.81429
15	0.0520094	$3e^{-5}$	16	9	0.64680	1.85639
16	0.198757	$2e^{-5}$	32	7	0.60388	2.15287
17	0.0623825	$5e^{-5}$	32	7	0.64436	1.82313
18	0.290875	$2e^{-5}$	32	5	0.57721	2.37405
19	0.118545	$5e^{-5}$	32	7	0.64588	1.81662
20	0.276562	$5e^{-5}$	32	9	**0.65101 ****	1.81706
21	0.15625	$2e^{-5}$	32	9	0.62046	2.02778
22	0.116603	$2e^{-5}$	32	5	0.57704	2.37062
23	0.176025	$3e^{-5}$	16	5	0.62156	1.97485

Table 5.3: Hyperparameter tuning results for KLDB ID classification on DZHW occupation data for approach 1. w_decay - weight decay, lr - learning rate, train_bs - batch size, val_acc - validation accuracy, val_loss - validation loss

samples from the custom dataset. We have used the following hyperparameter configuration weight decay - 0.0520094, learning rate - $3e^{-5}$, input batch size - 16, and 9 epochs for the KLDB ID classification and achieved similar results to the KLDB ID classification in approach 1.

Trial	w_decay	lr	train_bs	epochs	val_acc	val_loss
0	0.0550304	$5e^{-5}$	16	5	0.64142	1.83953
1	0.0468056	$4e^{-5}$	16	5	0.63334	1.88544
2	0.137775	$3e^{-5}$	16	9	0.64554	1.86463
3	0.00617535	$2e^{-5}$	32	9	0.61558	2.04153
4	0.0637017	$5e^{-5}$	32	7	0.64276	1.83923
5	0.185244	$2e^{-5}$	32	5	0.57157	2.39511
6	0.129584	$3e^{-5}$	32	5	0.59951	2.14601
7	0.119958	$5e^{-5}$	16	9	**0.65749 ***	1.83003
8	0.136821	$4e^{-5}$	32	9	0.63906	1.86255
9	0.114739	$5e^{-5}$	32	9	0.64579	1.83467
10	0.0139351	$4e^{-5}$	32	5	0.61945	2.00349
11	0.13515	$2e^{-5}$	16	5	0.60136	2.16629
12	0.28969	$2e^{-5}$	32	5	0.57022	2.39534
13	0.00478988	$2e^{-5}$	32	5	0.57199	2.39023
14	0.204979	$5e^{-5}$	32	9	**0.64848 *****	1.82402
15	0.0520094	$3e^{-5}$	16	9	0.64756	1.86575
16	0.198757	$2e^{-5}$	32	7	0.59665	2.17398
17	0.0623825	$5e^{-5}$	32	7	0.64142	1.85038
18	0.290875	$2e^{-5}$	32	5	0.56938	2.39486
19	0.118545	$5e^{-5}$	32	7	0.64175	1.83873
20	0.276562	$5e^{-5}$	32	9	**0.64983 ****	1.83522
21	0.15625	$2e^{-5}$	32	9	0.61415	2.04337
22	0.116603	$2e^{-5}$	32	5	0.57241	2.39325
23	0.176025	$3e^{-5}$	16	5	0.62358	1.98135

Table 5.4: Hyperparameter tuning results for KLDB ID classification on DZHW occupation data for approach 2. w_decay - weight decay, lr - learning rate, train_bs - batch size, val_acc - validation accuracy, val_loss - validation loss

In the next step, we performed hyperparameter tuning on the DZHW occupation dataset for the BERT model, which was fine-tuned on the above-mentioned relevance classification and KLDB ID classification tasks for the domain data. The results of the hyperparameter tuning step are mentioned in the Table 5.4. After this step, we chose the best configuration from the hyperparameter tuning step and performed a stratified 5-fold cross-validation step on the BERT model as mentioned in Section 5.3.1. The mean accuracy of the model on the test datasets from the stratified k-fold-cross-validation step was 66.68% with a standard deviation of 0.16. In addition to the above-mentioned steps, we increased the number of training epochs to 15 for Approaches 1 and 2 to observe its effects on the accuracy of test data. However, the test data accuracy did not increase with the increase in the training epochs.

Trial	w_decay	lr	train_bs	epochs	val_acc	val_loss
0	0.0550304	$5e^{-5}$	16	5	0.68963	1.63083
1	0.0468056	$4e^{-5}$	16	5	0.67633	1.73878
2	0.137775	$3e^{-5}$	16	9	**0.70475 ****	1.5972
3	0.00617535	$2e^{-5}$	32	9	0.65009	1.99733
4	0.0637017	$5e^{-5}$	32	7	0.23019	4.22415
5	0.185244	$2e^{-5}$	32	5	0.58242	2.57916
6	0.129584	$3e^{-5}$	32	5	0.61884	2.18926
7	0.119958	$5e^{-5}$	16	9	**0.71020 ***	1.5675
8	0.136821	$4e^{-5}$	32	9	0.70111	1.61009
9	0.114739	$5e^{-5}$	32	9	0.42324	3.08806
10	0.0139351	$4e^{-5}$	32	5	0.64980	1.9483
11	0.13515	$2e^{-5}$	16	5	0.62429	2.22524
12	0.28969	$2e^{-5}$	32	5	0.58184	2.58215
13	0.00478988	$2e^{-5}$	32	5	0.57966	2.56316
14	0.204979	$5e^{-5}$	32	9	0.68978	1.6915
15	0.0520094	$3e^{-5}$	16	9	**0.70417 *****	1.6056
16	0.198757	$2e^{-5}$	32	7	0.63097	2.22991
17	0.0623825	$5e^{-5}$	32	7	0.68781	1.68077
18	0.290875	$2e^{-5}$	32	5	0.58118	2.5784
19	0.118545	$5e^{-5}$	32	7	0.69872	1.64557
20	0.276562	$5e^{-5}$	32	9	0.70366	1.58451
21	0.15625	$2e^{-5}$	32	9	0.65758	2.00273
22	0.116603	$2e^{-5}$	32	5	0.58315	2.57891
23	0.176025	$3e^{-5}$	16	5	0.66201	1.91109

Table 5.5: Hyperparameter tuning results for KLDB ID classification on domain and DZHW occupation data for dataset dimension. w_decay - weight decay, lr - learning rate, train_bs - batch size, val_acc - validation accuracy, val_loss - validation loss

5.3.3 Dataset Dimension

We have appended the domain data in the dataset dimension with the DZHW data only during the training phase. So, we did the hyperparameter tuning of the 'gbert-base' model on the DZHW occupation data and domain data during the training step. It was ensured that the instances from the domain data were not present in test data during the stratified k fold-cross-validation to determine the performance of the model. The results for the hyperparameter tuning step can be observed in Table 5.5. In the cross-validation step, the BERT model achieved an accuracy of 67.25 % with a standard deviation of 0.04.

5.3.4 Baseline

As mentioned in Chapter 4, to compare the performance of the models from the training and dataset dimensions, we have implemented a baseline model, which is fine-tuned to perform the KLDB ID classification task only on the domain data. So we have not provided the domain data related to KLDB to the baseline 'gbert-base' model and performed the hyperparameter tuning (see Table 5.6) and stratified k-fold cross-validation steps. In the cross-validation step, the model attained an accuracy of 66.5 % with a standard deviation of 0.01.

5.4 Discussion

After performing the hyperparameter tuning and the cross-validation steps on for BERT models from the Training, and Dataset dimensions and the baseline we summarize the results of these models using the accuracy metric for the occupation coding task on the DZHW occupation data in the Table 5.7. As mentioned in the RQ-1, we provided the classification scheme's domain knowledge through the training dimension classification tasks. To evaluate whether providing the domain knowledge through fine-tuning BERT on domain data helps to improve the performance, as mentioned in RQ-2, we compare the accuracy of BERT from Approaches 1 and 2 with the baseline model. It can be observed that the performance of BERT with domain knowledge tends to be on a similar accuracy level to the baseline. But when the domain data was augmented with the DZHW occupation data as described in the Dataset dimension (RQ-3), the accuracy of the BERT model after the cross-validation step was higher than the baseline. Even though it was not a steep increase in the accuracy of the model, there was a minor improvement.

To further understand the performance of models on the various classes present in the DZHW occupation data, we considered the f1-score for all the 877 unique classes in the test data. Table 5.8 indicates the number of KLDB IDs with zero f1-score for all four BERT models. It can be observed that the number of KLDB IDs with zero f1-score slightly decreased in the case of BERT models from the training and dataset dimensions. Moreover, most classes with zero f1-score had less number of examples in the test and train data. The model with domain information ($\text{BERT}_{\text{Approach-1}}$) had a f1-score greater than 0.8 for more classes than the other BERT models.

Further analysis showed that the KLDB IDs in Table 5.9 had an f1-score greater than 0.8 for all the BERT models. It was observed that the common KLDB IDs do

Trial	w_decay	lr	train_bs	epochs	val_acc	val_loss
0	0.0550304	$5e^{-5}$	16	5	0.64091	1.82459
1	0.0468056	$4e^{-5}$	16	5	0.63502	1.88609
2	0.137775	$3e^{-5}$	16	9	0.64268	1.85129
3	0.00617535	$2e^{-5}$	32	9	0.61920	2.04203
4	0.0637017	$5e^{-5}$	32	7	0.64655	1.82689
5	0.185244	$2e^{-5}$	32	5	0.57258	2.39953
6	0.129584	$3e^{-5}$	32	5	0.60506	2.13539
7	0.119958	$5e^{-5}$	16	9	**0.65446 ***	1.83642
8	0.136821	$4e^{-5}$	32	9	0.64344	1.83659
9	0.114739	$5e^{-5}$	32	9	**0.64941 ****	1.81643
10	0.0139351	$4e^{-5}$	32	5	0.62080	2.00238
11	0.13515	$2e^{-5}$	16	5	0.60254	2.17286
12	0.28969	$2e^{-5}$	32	5	0.57081	2.39629
13	0.00478988	$2e^{-5}$	32	5	0.57401	2.39316
14	0.204979	$5e^{-5}$	32	9	0.64571	1.82294
15	0.0520094	$3e^{-5}$	16	9	0.64579	1.85534
16	0.198757	$2e^{-5}$	32	7	0.60195	2.17561
17	0.0623825	$5e^{-5}$	32	7	0.64057	1.83781
18	0.290875	$2e^{-5}$	32	5	0.57123	2.3976
19	0.118545	$5e^{-5}$	32	7	0.64360	1.825
20	0.276562	$5e^{-5}$	32	9	**0.6489 *****	1.81565
21	0.15625	$2e^{-5}$	32	9	0.61735	2.04139
22	0.116603	$2e^{-5}$	32	5	0.57401	2.3948
23	0.176025	$3e^{-5}$	16	5	0.6245	1.98062

Table 5.6: Hyperparameter tuning results for KLDB ID classification on DZHW occupation data for baseline. w_decay - weight decay, lr - learning rate, train_bs - batch size, val_acc - validation accuracy, val_loss - validation loss

Model	Accuracy
BERT fine-tuned on KLDB ID classification (Approach-1)	66.67 ± 0.11
BERT fine-tuned on relevance classification & KLDB ID classification (Approach-2)	66.68 ± 0.16
BERT with domain data during training phase	67.25 ± 0.04
BERT baseline	66.5 ± 0.01

Table 5.7: Summary of the cross-validation results for the BERT models on occupation coding task

Model	f1 - score			
	0.0	>0.0 and <= 0.5	>0.5 and <=0.8	>0.8
BERT $_{\text{Baseline}}$	406	178	226	67
BERT $_{\text{Approach-1}}$	402	181	213	80
BERT $_{\text{Approach-2}}$	403	175	229	70
BERT $_{\text{Dataset dimension}}$	400	185	220	72

Table 5.8: Performance of BERT models based on the distribution of f1-score for 877 classes

11293	33112	73324	81743
11294	54101	81102	81804
12104	62322	81112	81822
12144	63322	81142	82542
22342	71433	81234	83112
23322	71524	81353	83124
24232	72243	81454	83314
27212	72304	81474	84114
31114	73104	81504	84124
32122	73134	81624	84134
32142	73154	81733	84304

Table 5.9: Common KLDB IDs with f1-score > 0.8 for all the BERT models

not consist of IDs beginning with 0 and 4. Also, the KLDB IDs belonging to the Occupational field (Berufsbereiche) 8 - Gesundheit, Soziales, Lehre und Erziehung and Main occupational group (Berufshauptgruppenwere) 81 - Medizinische Gesundheitsberufe were more in the common KLDB IDs for f1-score > 0.8. Since BERT $_{\text{Approach-1}}$ had higher KLDB IDs with an f1-score higher than 0.8, we further compared its results with the BERT $_{\text{Baseline}}$ to examine the additional KLDB IDs on which BERT $_{\text{Approach-1}}$ performed better. Table 5.10 indicates the classes on which the BERT $_{\text{Approach-1}}$ had a better performance when compared with the baseline. In addition, BERT $_{\text{Approach-1}}$, BERT $_{\text{Dataset dimension}}$, and BERT $_{\text{Approach-2}}$ models had 54 KLDB IDs in common for the higher range of f1-score (> 0.8). On further investigation, we observed that BERT models struggled to classify the responses belonging to Militär and Naturwissenschaft, Geografie und Informatik occupational fields (KLDB ID - 0 and 4) and performed well, especially for Gesundheit, Soziales, Lehre und Erziehung (KLDB ID - 8) mentioned in Table 2.5. Table 5.11 depicts the occupation field wise distribution of higher f1-score for all the BERT models. We further represent the occupation field-wise distribution of the other f1-score ranges in the Table 5.12 and Table 5.13.

21112	34302	61323	84414
23422	41184	72302	84513
24422	41393	73124	91344
27184	42124	81342	92122
27223	52413	82284	93222
31214	52132	84214	94214

Table 5.10: KLDB IDs with f1-score > 0.8 for BERT $_{\text{Approach-1}}$ vs BERT $_{\text{Baseline}}$

KLDB ID	Baseline	Approach-1	Approach-2	Dataset dimension
0	1	0	0	0
1	5	5	5	5
2	6	9	7	7
3	5	6	4	4
4	0	3	1	2
5	4	6	4	5
6	3	3	5	6
7	10	11	10	11
8	25	28	26	25
9	8	9	8	7

Table 5.11: Distribution of f1-score > 0.8 for level-1 KLDB IDs (occupational fields) among the BERT models

5.5 Summary

In this chapter, we discussed the initial experiments performed on the custom dataset created from the KLDB and the DZHW occupation dataset. We presented the hyperparameter tuning results of the BERT models for the occupation coding tasks. The hyperparameter tuning was performed on the custom dataset and the DZHW datasets. Further, we discussed and implemented the stratified k-fold cross-validation step and chose the accuracy metric to evaluate the performance of the BERT models for the occupation coding task. We observe that the models trained on domain data had shown minimal improvement in terms of accuracy when compared with the baseline. Further, we examined the results using classification report [6] and considered f1-score to analyze the performance of models for the 877 classes. We observed that all the BERT models did not perform well in particularly classifying more than 400 classes due to the fewer training examples and lack of text in user responses. We further examined the results for the f1-score ranges and presented the class-wise distribution of f1-scores.

[6] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

KLDB ID	Baseline	Approach-1	Approach-2	Dataset dimension
0	0	0	0	0
1	10	11	13	12
2	40	37	41	40
3	13	12	13	12
4	21	21	24	22
5	15	15	17	15
6	20	20	24	22
7	33	28	28	27
8	48	45	44	46
9	26	24	25	24

Table 5.12: Distribution of f1-score > 0.5 and $<= 0.8$ for level-1 KLDB IDs (occupational fields) among the BERT models

KLDB ID	Baseline	Approach-1	Approach-2	Dataset dimension
0	0	0	0	0
1	7	9	6	9
2	26	27	24	27
3	7	9	8	9
4	22	21	23	22
5	16	14	14	15
6	20	21	18	20
7	35	38	35	37
8	29	28	30	30
9	16	14	17	16

Table 5.13: Distribution of f1-score > 0.0 and $<= 0.5$ for level-1 KLDB IDs (occupational fields) among the BERT models

6. Conclusion and Future Work

6.1 Conclusion

Occupation coding is considered an essential step in Socio-Economic studies as it helps standardize the user responses collected from various occupation-related surveys and further analyze the standardized responses. The occupation coding task often relies upon official classification schemes to classify or categorize the survey respondents' occupations. During the literature review, we observed that occupation coding methods are categorized into manual and (semi-) automatic methods. The (semi-) automatic methods are further divided into rule-based and Machine Learning based methods. In Machine Learning, occupation coding is considered a text classification problem. However, it has specific challenges as the text in the user responses is often very short with keywords and contains many target classes for the classification. In the literature review, we further observed that the official occupation classification schemes were not utilized with the pre-trained language models to perform occupation coding.

Hence, we decided to use the domain knowledge provided in the official occupation schemes to tackle the short text scenario and utilize the pre-trained language models (BERT) due to their capability to capture the semantics from the text information. So we proposed an approach to integrate the domain knowledge from the KLDB classification scheme through additional classification tasks and then perform the occupation coding on the anonymized responses from DZHW occupation data. We proposed to use relevance classification, and occupation coding classification tasks to impart domain knowledge into BERT based on [BB21] and [YSL19]. For the additional classification tasks, we created custom datasets from the KLDB classification scheme and fine-tuned BERT on these custom datasets before fine-tuning on the anonymized DZHW occupation data. In addition, we proposed to augment the domain data with the DZHW occupation data during the training phase as an additional approach to integrate the domain knowledge.

Furthermore, we performed experiments related to hyperparameter tuning and cross-validation for the proposed approach and used a baseline to evaluate whether in-

tegrating the domain knowledge improves the classification performance of BERT on the DZHW occupation data. After the cross-validation steps, we compared the accuracy of the BERT models from the proposed approach with the baseline model, we observed that there wasn't a drastic increase in the accuracy of the BERT models with the domain knowledge, but there was a slight improvement. To further examine the performance of BERT models on the various classes in DZHW data, we observed the class-wise f1-score distribution. We observed that BERT could not classify approximately 400 of the 877 classes due to the lack of enough text and enough training examples. In addition, the models with domain knowledge had more classes with f1-score > 0.8 compared to the baseline BERT with no domain information.

6.2 Future Work

In this section, we discuss the scope for improvements and future work that was identified during the thesis.

Including additional occupation titles: The KLDB classification scheme also provides information about the additional job titles that are included under a KLDB ID. The relevant activities for these job titles are not provided in the KLDB scheme since it provides generalized information about the activities for a KLDB ID. So a semantic similarity-based approach can be used to identify relevant activities for the additional job titles, and this information can be added to the custom dataset for the relevance classification task to fine-tune BERT on the domain data, or it could be added to the DZHW occupation data during the training phase of the BERT model.

Information retrieval approach: Instead of automatically classifying the user response by a Machine Learning or Deep Learning model, the occupation coding task can be converted into an information retrieval problem, where the system suggests a list of suitable KLDB IDs for user responses. The user responses and the occupation information provided in the KLDB scheme can be converted into embeddings using a pre-trained model like SBERT. Then similarity between the responses and occupation embeddings can be computed, and the most suitable KLDB IDs based on a ranking mechanism can be suggested. Further, this approach can be evaluated through metrics like precision and recall since the DZHW data consists of KLDB ID assigned to the user responses.

A. Appendix

Trial	w_decay	lr	train_bs	epochs	val_acc	val_loss
0	0.28185	$3.6e^{-5}$	25	8	0.641	1.8521
1	0.179055	$3,00e^{-5}$	32	4	0.58419	2.2867
2	0.0174251	$2,00e^{-5}$	16	7	0.62400	1.9989
3	0.268448	$2.4e^{-5}$	19	5	0.60969	2.1127
4	0.109527	$3.6e^{-5}$	32	4	0.60018	2.1659
5	0.0748107	$3,00e^{-5}$	38	8	0.62837	1.9595
6	0.0912727	$3,00e^{-5}$	16	4	0.61120	2.0783
7	0.109527	$3,00e^{-5}$	19	4	0.60489	2.1157
8	0.33822	$2,00e^{-5}$	30	8	0.61062	2.0880
9	0.0698314	$3,00e^{-5}$	32	8	0.63275	1.9311
10	0.224196	$3.6e^{-5}$	16	8	0.64739	1.8365
11	0.171133	$2,00e^{-5}$	19	4	0.57392	2.3635
12	0.0975991	$2,00e^{-5}$	9	4	0.60388	2.1608
13	0.282661	$3,00e^{-5}$	32	8	0.62905	1.9358
14	0.109527	$3.6e^{-5}$	12	4	0.62400	1.9554
15	0.33822	$2.88e^{-5}$	30	8	0.62980	1.9402
16	0.290875	$3.6e^{-5}$	32	4	0.59782	2.1974
17	0.117318	$3,00e^{-5}$	16	5	0.62315	1.9855
18	0.0935133	$3,00e^{-5}$	32	8	0.63266	1.9289
19	0.22548	$4.32e^{-5}$	30	8	0.64091	1.8449

Table A.1: Hyperparameter tuning results of baseline BERT model for the KLDB ID classification on DZHW occupation data using PBT method. w_decay - weight decay, lr - learning rate, train_bs - batch size, val_acc - validation accuracy, val_loss - validation loss.

B. Abbreviations and Notations

Acronym	Meaning
ALLBUS	Die Allgemeine Bevölkerungsumfrage der Sozialwissenschaften
KLDB	Klassification der Berufe
ISCO	International Standard Classification of Occupations
BA	Bundesagentur für Arbeit
DZHW	Deutsche Zentrum für Hochschul- und Wissenschaftsforschung
ILO	International Labour Organization
NOC	National Occupation Classification
ML	Machine Learning
DL	Deep Learning
QA	Question-Answering
ASR	Automatic Speech Recognition
NLI	Natural Language Inference
STS	Semantic Textual Similarity
NLP	Natural Language Processing
ReLU	Rectified Linear Unit
FFNN	Feed forward Neural Network
CNN	Convolution Neural Network
RNN	Recurrent Neural Network
LSTM	Long short-term Memory
ELMO	Embeddings from Language Model
BERT	Bidirectional Encoder Representations from Transformers
SBERT	Sentence BERT
GPT	Generative Pre-trained Transformer
MLM	Masked Language Modeling
NSP	Next Sentence Prediction
TF – IDF	Term Frequency - Inverse Document Frequency
CRF	Conditional Random Field

Bibliography

[BB21] Alexander Brinkmann and Christian Bizer. Improving hierarchical product classification using domain-specific language modelling. volume 44, pages 14–25, 2021. (cited on Page 25, 29, and 49)

[BBA20] Hongchang Bao, Christopher J O Baker, and Anil Adisesh. Occupation coding of job titles: Iterative development of an automated coding algorithm for the canadian national occupation classification (aca-noc). 2020. (cited on Page 2, 3, and 22)

[BBKP14] Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Padó. GermEval 2014 Named Entity Recognition Shared Task: Companion Paper. In *Proceedings of the KONVENS GermEval workshop*, pages 104–112, 2014. (cited on Page 18)

[BCB15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, 2015. (cited on Page 14)

[Bie20] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com. (cited on Page 35)

[BWC+21] Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, J. Yu, and Yunhai Tong. Syntax-bert: Improving pre-trained transformers with syntax trees. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2021. (cited on Page 24)

[CGL16] Roxanne Connelly, Vernon Gayle, and Paul S. Lambert. A review of occupation-based social classifications for social survey research. volume 9, 2016. (cited on Page 1)

[CMSW] Robert H. Creecy, Brij M. Masand, Stephen J. Smith, and David L. Waltz. Trading mips and memory for knowledge engineering. volume 35, page 48–64, New York, USA. Association for Computing Machinery, 1992. (cited on Page 23)

[Con97] Frederick Conrad. *Using Expert Systems to Model and Improve Survey Classification Processes*, chapter 17, pages 393–414. John Wiley Sons, Ltd, 1997. (cited on Page 21)

[CSM] Branden Chan, Stefan Schweter, and Timo Möller. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain. International Committee on Computational Linguistics, 2020. (cited on Page 18)

[DCLT] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics, 2019. (cited on Page ix, 16, 17, 18, and 37)

[Dec21] Decorte, Jens-Joris and Van Hautte, Jeroen and Demeester, Thomas and Develder, Chris. JobBERT : understanding job titles through skills. In *FEAST, ECML-PKDD Workshop*, page 9, 2021. (cited on Page 23)

[EPR14] Birch M. Elias P. and Ellison R. Cascot international version 5," user guide [online], Institute for Employment Research, University of Warwick, Coventry, 2014. (cited on Page 21, 22, and 23)

[fS19] GESIS Leibniz-Institut für Sozialwissenschaften. Allbus/ggss 2018 (allgemeine bevölkerungsumfrage der sozialwissenschaften/german general social survey 2018). GESIS Data Archive, Cologne. ZA5270 Data file Version 2.0.0, https://doi.org/10.4232/1.13250, 2019. (cited on Page 1)

[GLT21] Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention in natural language processing. volume 32, pages 4291–4308. Institute of Electrical and Electronics Engineers (IEEE), 2021. (cited on Page 14)

[GSK+17] Hyukjun Gweon, Matthias Schonlau, Lars Kaczmirek, Michael Blohm, and Stefan Steiner. Three methods for occupation coding based on statistical learning. volume 33, pages 101–122. Statistics Sweden (SCB), 2017. (cited on Page 23)

[HR] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 328–339, Melbourne, Australia. Association for Computational Linguistics, 2018. (cited on Page 16)

[HS] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. volume 9, page 1735–1780, Cambridge, MA, USA. MIT Press, 1997. (cited on Page 12)

[HTS12] Josef Hartmann, Nikolai Tschersich, and Gerd Schütz. Die vercodung der offenen angaben zur beruflichen tätigkeit nach der klassifikation der berufe 2010 (KldB 2010) und nach der international standard classification of occupations 2008 (ISCO 08). 2012. (cited on Page 21)

[HW12] W Hacking and L Willenborg. Coding; interpreting short descriptions using a classification. 2012. (cited on Page 22)

[HZG03] Jürgen H. P. Hoffmeyer-Zlotnik and Alfons J. Geis. Berufsklassifikation und messung des beruflichen status/ prestige. volume 27, pages 125–138, 2003. (cited on Page 21)

[ILSW20] Akina Ikudo, Julia I. Lane, Joseph Staudt, and Bruce A. Weinberg. Occupational classifications: A machine learning approach. volume 44, pages 57–87. IOS Press, 2020. (cited on Page 22)

[ISC08] ISCO-08. International Standard Classification of Occupations Structure, group definitions and correspondence tables. volume 1, 2008. (cited on Page 5 and 7)

[Kim] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics, 2014. (cited on Page 8)

[KJMH+19] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. volume 10, 2019. (cited on Page 22)

[KPBP] Anna Koufakou, Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. HurtBERT: Incorporating lexical features with BERT for the detection of abusive language. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 34–43, Online. Association for Computational Linguistics, 2020. (cited on Page 24)

[KZS15] Gregory R. Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015. (cited on Page 18)

[LLN+18] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. volume abs/1807.05118, 2018. (cited on Page 35)

[LPGL] Gaël Letarte, Frédérik Paradis, Philippe Giguère, and François Laviolette. Importance of self-attention for sentiment analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 267–275, Brussels, Belgium. Association for Computational Linguistics, 2018. (cited on Page 14)

[LTM] Wah Meng Lim and Harish Tayyar Madabushi. UoB at SemEval-2020 task 12: Boosting BERT with corpus level information. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2216–2221, Barcelona (online). International Committee for Computational Linguistics, 2020. (cited on Page 23)

[MCBB09] Leslie A. MacDonald, Alex Cohen, Sherry Baron, and Cecil M. Burchfiel. Occupation as Socioeconomic Status or Environmental Exposure? A Survey of Practice Among Population-based Cardiovascular Studies in the United States. volume 169, pages 1411–1421, 2009. (cited on Page 1)

[Mea14] Alexander Measure. Automated coding of worker injury narratives. 2014. (cited on Page 22)

[MK03] A 't Mannetje and H Kromhout. The use of occupation and industry classifications in general population studies. volume 32, pages 419–428, 2003. (cited on Page 1)

[MKC+] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: A comprehensive review. volume 54, New York, NY, USA. Association for Computing Machinery, 2021. (cited on Page 22)

[MP18] Marcin Michał Mirończuk and Jarosław Protasiewicz. A recent overview of the state-of-the-art elements of text classification. volume 106, pages 36–54, 2018. (cited on Page 22)

[NTTS20] Amber Nigam, Shikha Tyagi, Kuldeep Tyagi, and Arpan Saxena. Skillbert:"skilling" the bert to classify skills! 2020. (cited on Page 23)

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Curran Associates Inc., Red Hook, NY, USA, 2019. (cited on Page 35)

[PM13] Wiebke Paulus and Britta Matthes. The German classification of occupations 2010 : structure, coding and conversion table. Technical report, 2013. (cited on Page 2, 7, 8, and 27)

[PMB12] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. volume abs/1211.5063. CoRR, 2012. (cited on Page 12)

[PNI+] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 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, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics, 2018. (cited on Page 16)

[PP13] Sangameshwar Patil and Girish K Palshikar. Surveycoder: A system for classification of survey responses. In *International Conference on Application of Natural Language to Information Systems*, pages 417–420. Springer, 2013. (cited on Page 22)

[PTM] Anushka Prakash and Harish Tayyar Madabushi. Incorporating count-based features into pre-trained models for improved stance detection. In *Proceedings of the 3rd NLP4IF Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 22–32, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL), 2020. (cited on Page 24)

[RG] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics, 2019. (cited on Page ix, 16, 18, 19, and 33)

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. volume 323, pages 533–536. Nature Publishing Group, 1986. (cited on Page 12)

[RN18] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. (cited on Page 16)

[RSR+19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. volume abs/1910.10683, 2019. (cited on Page 16)

[RZLL] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics, 2016. (cited on Page 17)

[Sch14] Malte Schierholz. Automating survey coding for occupation. FDZ-Methodenreport 201410 (en), Institut für Arbeitsmarkt- und Berufsforschung (IAB), Nürnberg [Institute for Employment Research, Nuremberg, Germany], 2014. (cited on Page 21 and 22)

[SS20] Malte Schierholz and Matthias Schonlau. Machine Learning for Occupation Coding—A Comparison Study. volume 9, pages 1013–1034, 2020. (cited on Page 3, 22, and 23)

[SSW+19] Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Shijing Si, Dinghan Shen, Dong Wang, and Lawrence Carin. Syntax-infused transformer and bert models for machine translation and natural language understanding. 2019. (cited on Page 24)

[Ste11] Max Friedrich Steinhardt. The wage impact of immigration in germany - new evidence for skill groups and occupations. volume 11, 2011. (cited on Page 1)

[TTTL14] Kazuko Takahashi, Hirofumi Taki, Shunsuke Tanabe, and Wei Li. An automatic coding system with a three-grade confidence level corresponding to the national/international occupation and industry standard open to the public on the web. 2014. (cited on Page 23)

[TVL21] Hieu Tran, Hanh Hong-Phuc Vo, and Son T. Luu. Predicting job titles from job descriptions with multi-label text classification. pages 513–518, 2021. (cited on Page 23)

[VSP+] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc., 2017. (cited on Page ix, 8, 14, 15, 16, and 17)

[WDS+] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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. Association for Computational Linguistics, 2020. (cited on Page 16 and 35)

[WNB] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics, 2018. (cited on Page 17)

[WS18] Michael Wiegand and Melanie Siegel. Overview of the germeval 2018 shared task on the identification of offensive language. 2018. (cited on Page 18)

[YKYS17] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. volume abs/1702.01923, 2017. (cited on Page 8)

[YSL19] Shanshan Yu, Jindian Su, and Da Luo. Improving bert-based text classification with auxiliary sentence and domain knowledge. volume 7, pages 176600–176612. IEEE, 2019. (cited on Page 29 and 49)

[YTM] Erik Yan and Harish Tayyar Madabushi. UoB at SemEval-2021 task 5: Extending pre-trained language models to include task and domain-specific information for toxic span prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 243–248. Association for Computational Linguistics, 2021. (cited on Page 24 and 25)

[ZWZ+20] Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635, 2020. (cited on Page 24)

Declaration of Academic Integrity

I hereby declare that I have written the present work myself and did not use any sources or tools other than the ones indicated.

Datum: .....

(Signature)